ARL-TR-8074 ● AUG 2017

US Army Research Laboratory

Quantifying Uncertainty from Computational Factors in Simulations of a Model Ballistic System

by Daniel J Hornbaker

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-8074 ● AUG 2017

US Army Research Laboratory

Quantifying Uncertainty from Computational Factors in Simulations of a Model Ballistic System

by Daniel J Hornbaker

Weapons and Materials Research Directorate, ARL

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* August 2017	**2. REPORT TYPE** Technical Report	**3. DATES COVERED (From - To)** November 2015–November 2016

4. TITLE AND SUBTITLE Quantifying Uncertainty from Computational Factors in Simulations of a Model Ballistic System	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Daniel J Hornbaker	**5d. PROJECT NUMBER** 1L162618AH80
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-WMP-E Aberdeen Proving Ground, MD 21005-5066	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-8074
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Uncertainty in simulations of a model ballistic system is studied, with a focus on factors that are primarily related to the numerical structuring of a problem, such as cell size, domain extent, and system orientation. Depth of penetration of a threat into a target is tracked as the primary quantity of interest. Overall, computational time serves as a secondary performance metric. Two different simulation codes developed at Sandia National Laboratories, CTH and ALEGRA, are employed in this investigation. This study provides a foundation for future explorations of uncertainty in ballistic simulations, as well as a template for investigation of uncertainty in simulations of other systems of interest.

15. SUBJECT TERMS

numerical simulation, ballistic, uncertainty quantification, CTH, ALEGRA

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Daniel J Hornbaker
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	66	**19b. TELEPHONE NUMBER (Include area code)** 410-278-7697

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank Stephen Schraml, Robert Doney, and Steven Segletes for undertaking a thorough review of this work. I would also like to acknowledge the invaluable ongoing collaboration between the US Army Research Laboratory and the CTH and ALEGRA development teams at Sandia National Laboratories. I would particularly like to thank Duane Labreche for his assistance in developing the ALEGRA version of the model ballistic system utilized in this report.

This work was supported in part by a grant of computer time from the Department of Defense High Performance Computing Modernization Program at the US Army Research Laboratory Department of Defense Supercomputing Resource Center.

1. Introduction

Scientific investigation can be broadly grouped into 3 domains: experimental, theoretical, and computational. Experimental science involves direct tests and observations of phenomena in the real world. Theoretical science seeks to formulate sweeping explanations that account for both existing data and future possibilities.

Computational science is a relatively new mode of investigation, with less than a century since Turing's pioneering formulation of automated computing.[1] Modern computational science involves the use of digital computers to solve mathematical models of various phenomena. Whereas application of theoretical science has historically been limited by the human capacity for complexity and ability to perform long calculations, computational methods allow encoding of numerous theoretical equations and empirical data, and the execution of billions of mathematical operations, to compute outcomes for systems of interest.

Computational science is a powerful method of investigation, contributing to major advancements across a wide array of subjects including climatology, medicine, economics, aerospace, and cosmology. To best utilize the tools of computational science, it is important to understand how these sophisticated calculations relate to phenomena in the real world. Numerical simulations are generally deterministic in nature; while stochastic methods exist, most rely on pseudorandom numbers, which use seeded algorithms to predictably generate sequences of numbers. Thus, repeated instances of a given simulation will produce the same result.

A foundational assumption of modern science is that the physical world operates objectively in a repeatable and predictable manner. Even in the quantum realm, where individual measurement outcomes are truly probabilistic, the distribution of outcomes is inherently fixed, and aggregated ensembles of particles behave in predictable ways. In spite of this, our ability to precisely measure or control any given quantity is limited, creating unavoidable variation in experimental outcomes. Repeated instances of the same nominal experiment will generally produce different results.

Strict repeatability is one fundamental difference between experiments and numerical simulations. Another is that while control over experimental parameters is limited and imprecise, numerical simulations require the user to assign concrete values to all inputs, regardless of whether appropriate values are accurately and precisely known. These differences highlight the importance of variability in the proper application of computational methods to problems in the real world. The study of the uncertainties associated with numerical simulations and experiments, and the connections between the two, is known as "uncertainty quantification".[2,3]

Computational science plays a significant role in the development of ballistic protection technologies within the US Department of Defense (DOD). It is common practice to employ large-scale physics-based numerical simulations in the design and evaluation of protection systems. Compared with traditional experimental approaches, simulations are often faster, less expensive, and offer greater control over system conditions. They also enable detailed representations of complex ballistic interactions that can be impossible to observe directly.

Ballistic simulations come in countless varieties and can involve a vast number of parameters. In this investigation, a simple model system is employed to study the influence of various numerical factors, such as cell size, domain extent, and system orientation, on simulation outcomes.

Even within this limited scope, a thorough investigation employing full design-of-experiment methodology[4] would be daunting. Instead, this work relies on traditional single-parameter variation to quantify outcome sensitivity to simulation parameters. In addition, this study highlights the influence that often-overlooked factors can have on simulation results and provide a template for others interested in pursuing similar investigations in their own areas of interest. In a more practical vein, this work provides an independent evaluation of how 2 multiphysics simulation codes widely used in the defense community, CTH and ALEGRA, respond to variations in problem structure for simulations involving a principal type of ballistic event. More than 200 simulations were completed during this research, with nearly 4 million core-hours of computing time invested.

A comprehensive review of the computational details in this work begins in the next section. The model ballistic system is then described, followed by a discussion of the metrics used to quantify simulation outcomes. Results for the various parameter studies are grouped along 4 general themes: simulation execution, domain structure, time-step controls, and physical invariance. This report concludes with a concise summary and discussion of directions for future research.

2. Computational Setup

One of the first factors decided in any simulation is the choice of simulation code. Two different multiphysics simulation codes were selected for study: CTH[5] version 11.1 and ALEGRA[6] release 21 May 2015. Both are US Department of Energy codes developed at Sandia National Laboratories, and both are capable of modeling the solid dynamics and shock physics of multiple deformable materials in up to 3 spatial dimensions. These codes belong to a general class of software known as hydrocodes.[7] Originating in the late 1950s and early 1960s to simulate high-rate impacts, the name derives from the approach at that time of approximating material in ballistic impacts as being essentially fluid, or hydrodynamic.

Another factor typically fixed at the outset is the computing hardware. Most simulations in this work were performed on an IBM iDataPlex supercomputer, with hostname Hercules, located at the US Army Research Laboratory (ARL) and managed by its DOD Supercomputing Resource Center. Hercules has 1092 compute nodes with sixteen 2.6-GHz cores and 64 GB of memory each. A few simulations were performed on other platforms for comparison. One platform was a second IBM iDataPlex supercomputer with hostname Pershing. Pershing has 1260 compute nodes with specifications identical to those in Hercules. A Cray XC40 supercomputer with hostname Excalibur was also used. Excalibur sports 3098 compute nodes having thirty-two 2.3-GHz cores and 128-GB memory each.

Instructions for simulations are provided to the codes in the form of text files. CTH expects input expressed in centimeter–gram–second (CGS) units by default, with temperature specified in electron-volts. ALEGRA was configured to use CGS units as well, though temperature in this code is specified in kelvins. Input files were written using the APREPRO algebraic preprocessing language[8] to simplify the process of varying simulation parameters. Examples of partial CTH and ALEGRA input files are provided in Appendixes A and B, respectively.

All simulations employed uniform 3-D rectilinear domains consisting of cubic cells. Resolution refers to the length of a cell edge; a domain consisting of 0.05- × 0.05- × 0.05-cm cubic cells has a resolution of 0.05 cm. In a quirk of terminology, decreasing the cell size is said to increase the resolution of a simulation, in that objects in the domain are more finely resolved. Domain boundaries in these simulations were typically free void, with pressure fixed to zero and mass allowed to flow out of (but not into) the domain. An exception to this was when symmetry planes were used, as discussed in Section 5.2.

A mixed Lagrangian–Eulerian computational scheme is used in both CTH and ALEGRA. In the Lagrange portion of each time step, the domain distorts in response to physical forces and then in the Euler portion the resulting material state is mathematically mapped back onto the original undistorted grid.

No material discards were used. Discarding troublesome material is typical, and often necessary, in simulations of ballistic systems in order to get simulations to run to completion. The usual culprits are small subcell fragments of material with physical states exceeding the limits of validity for the material models employed.

For example, it does not take much energy to cause a miniscule mass to reach a temperature of thousands of degrees kelvin, which is not necessarily wrong, as localized heating can occur. However, while materials in the real world respond naturally to high temperatures with phase changes and state excitations, simulations often lack models for these extreme behaviors since their impact on quantities of interest are often negligible and the computational cost of the calculations can be high.

Both codes have an array of parameters available for the user to customize the operation of the code. While the default values are suitable for many problems, in this work many of these parameters were explicitly assigned in the input files to provide a thorough accounting of code configuration. A brief description of some of these parameters for each code is provided in the following subsections. This information will be of interest mainly to users of these codes; other readers can proceed directly to Section 3.

2.1 CTH Configuration

The default MMP0 option was used to model thermodynamics in cells containing multiple materials. This option partitions volume changes and work energy in proportion to the volume fractions of materials in a cell.

The TBAD parameter was set to 1e30 to allow simulations to continue regardless of the number cells with potentially unrealistic thermodynamics states encountered.

Updated fracture logic was enabled by setting FRAC = 1.

The default energy convection control, in which internal energy is conserved and any resulting discrepancies in kinetic energy are discarded, was selected by setting CONVECTION = 0.

The Sandia Modified Youngs' Reconstruction Algorithm was employed for tracking material interfaces. A special fragment-moving model is implemented in CTH to handle motion of subcell material fragments embedded within a different material. This model was disabled for fragments of void material by setting NOFRAGMENT = 0.

A zero-velocity threshold value of 0.001 cm/s was used. Material with velocity less than this value had velocity changed to be exactly 0 cm/s.

Time steps were set to 0.55 times the Courant stability limit calculated by the code. The maximum allowed time-step ratio for subsequent cycles was capped at 1.068.

2.2 ALEGRA Configuration

IGNORE KINEMATIC ERRORS was enabled with the default limit on the stretch tensor eigenvalue (RESET EIGENVALUE = le-5). This is an error-handling scheme for dealing with various numerical problems that can arise in computing the stretch tensor.

PISCES HOURGLASS CONTROL was used with VISCOSITY = 0.05 to stiffen the zero-energy deformation modes of domain cells.

INTERNAL ENERGY ADVECTION was used, in which internal energy is conserved during remap and errors in kinetic energy are discarded.

The Sandia Modified Youngs' Reconstruction Algorithm was used to track material interfaces. The material advection scheme was set to the default MODERATE ADVECTION, which implements a third-order advection method for cells containing only a single material, with various alternatives for cells containing multiple materials. In ALEGRA, momentum is a node-centered quantity that uses a separate advection method from that used for cell-centered quantities. The default Half-Interval Shift method[9] was used for this.

The ISENTROPIC MULTIMATERIAL ALGORITHM was used to determine state variables in multimaterial cells, with the PRESSURE RELAXATION and TEMPERATURE RELAXATION algorithms turned on and THERMAL EQUILIBRIUM set to off.

Like CTH, ALEGRA computes a maximum stable time step based on numerical constraints imposed by the physics of the problem. In addition, a maximum allowed change in cell volume during the domain deformation phase was imposed by setting MAXIMUM VOLUME CHANGE = 0.5. By default, time steps are set to 0.9 of the calculated maximum (TIME STEP SCALE = 0.9). The maximum allowed time-step ratio for subsequent cycles was capped at 1.068.

3. Model Ballistic System

The model ballistic system selected for this study consists of a long rod projectile impacting a 15-cm-thick solid cubic block (Fig. 1). This system represents a common type of event important to armor design and is the sort of problem that hydrocodes were initially developed to model.

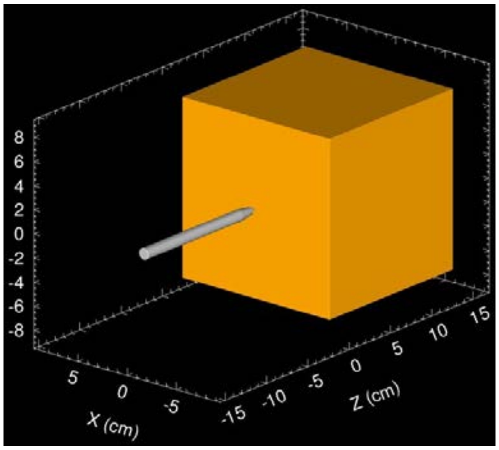

Fig. 1 Perspective rendering of the model ballistic system in CTH

The threat projectile is based on the 131W laboratory round commonly employed at ARL (Fig. 2).[10] The 131W is fabricated from 93% tungsten heavy alloy (WHA), which is composed of tungsten particles (93% by weight) sintered together in a matrix composed of 5% nickel and 2% iron.

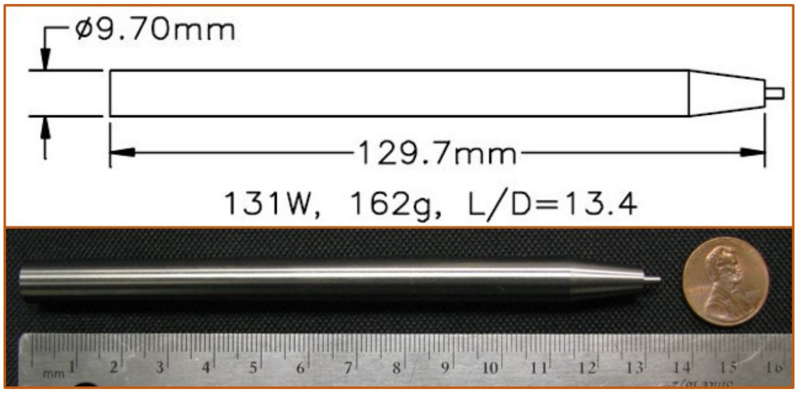

Fig. 2 Schematic (top) and photograph (bottom) of the 131W

The model system employs 2 materials commonly encountered in ballistic applications: WHA in the projectile and rolled homogeneous armor (RHA) steel[11] in the target. The physics of these materials are simulated using a combination of material models. An equation-of-state (EOS) model computes the thermodynamic properties of the material. A constitutive model generates an appropriate stress response to material deformation. A failure model is used to determine breakdowns in the integrity of a material.

Variables relating to the thermodynamic state of a material are mutually interdependent quantities. Both codes implement a Mie–Grüneisen EOS model,[12] in which the pressure P, density ρ, and specific internal energy E are related by

$$P = \rho^2 \frac{dU(\rho)}{d\rho} + \left(\frac{\partial P(\rho,E)}{\partial E}\right)_\rho [E - U(\rho)] \,. \tag{1}$$

U is the specific interatomic potential energy. The terms involving *U* can be eliminated if the thermodynamic states of the material are known along some reference curve. Where shock physics are involved, such as in ballistic impacts, the Hugoniot relations provide just such a reference.[13] A Hugoniot curve is the set of thermodynamic states (P_h, ρ_h, E_h) achievable from a given initial state (P_0, ρ_0, E_0) via a shock process. For a Hugoniot state having the same density as a state of interest ($\rho_h = \rho$), we have

$$P_h = \rho^2 \frac{dU(\rho)}{d\rho} + \left(\frac{\partial P(\rho,E)}{\partial E}\right)_\rho [E_h - U(\rho)] \,. \tag{2}$$

Subtracting Eq. 2 from Eq. 1 eliminates the terms involving *U*:

$$P - P_h = \left(\frac{\partial P(\rho,E)}{\partial E}\right)_\rho [E - E_h] \,. \tag{3}$$

The differential term in Eq. 3, when divided by density, is referred to as the Grüneisen parameter Γ[14]:

$$\Gamma(\rho) \equiv \frac{1}{\rho}\left(\frac{\partial P(\rho,E)}{\partial E}\right)_\rho = \frac{\alpha_v}{\kappa C_v \rho} \,. \tag{4}$$

The right-hand side of Eq. 4 expresses the Grüneisen parameter in terms of various measurable material properties: the volumetric thermal expansion coefficient α_v, the isothermal compressibility κ, and the specific heat capacity C_v. For many materials these properties are often treated as being independent of density, which then implies $\Gamma(\rho)\rho = \Gamma(\rho_0)\rho_0 \equiv \Gamma_0\rho_0$. Substituting into Eq. 3 gives

$$P - P_h = \Gamma_0 \rho_0 [E - E_h]. \tag{5}$$

Expressions for the Hugoniot state variables can be generated using the following conservation equations for material properties across a shock front[13]:

$$\text{Mass:} \quad \frac{\rho}{\rho_0} = \frac{u_s}{u_s - u_p} \tag{6}$$

$$\text{Momentum:} \quad P_h - P_0 = \rho_0 u_s u_p \tag{7}$$

$$\text{Energy:} \quad E_h - E_0 = \frac{1}{2}[P_h + P_0]\left[\frac{1}{\rho_0} - \frac{1}{\rho}\right] . \tag{8}$$

The quantities u_s and u_p are the shock front velocity and material velocity, respectively. A polynomial function is often fit to experimental measurements of these velocities:

$$u_s = C_s + s_1 u_p + \frac{s_2}{C_s} u_p^2 \,. \tag{9}$$

The quantity C_s is the bulk sound speed of the material. In the linear case ($s_2 = 0$), the system of Eqs. 5–9 can be combined to form the equation:

$$P = P_0(1 - \Gamma_0\chi) + \frac{\rho_0 C_s^2 \chi}{[1-\chi s_1]^2}\left(1 - \frac{\Gamma_0}{2}\chi\right) + \Gamma_0\rho_0[E - E_0];\;\; \chi = 1 - \frac{\rho_0}{\rho}\,. \qquad (10)$$

In materials undergoing shock, the pressure P typically becomes many orders of magnitude larger than the initial pressure P_0, and the leading term in Eq. 10 can be omitted. As a final step, if we take the specific heat capacity C_v to be roughly constant across the temperature range of the system, the change in energy can be related to a corresponding change in temperature T:

$$P = \frac{\rho_0 C_s^2 \chi}{[1-\chi s_1]^2}\left(1 - \frac{\Gamma_0}{2}\chi\right) + \Gamma_0\rho_0 C_v[T - T_0]\,. \qquad (11)$$

The Johnson–Cook constitutive equation is used to model the viscoplastic response of materials.[15] This is an empirical formula that relates the flow stress σ_y generated under plastic deformation to the plastic strain ε_p, and includes effects such as strain hardening, strain-rate hardening, and thermal softening:

$$\sigma_y = \left[A + B\varepsilon_p^N\right]\left[1 + C\ln\frac{\dot{\varepsilon}_p}{\dot{\varepsilon}_{p0}}\right]\left[1 - \left(\frac{T-T_0}{T_m-T_0}\right)^M\right]. \qquad (12)$$

A is the initial yield strength, while B and N relate to the strain-induced hardening. These parameters are generally determined from mechanical tensile or torsion data acquired in the quasi-static limit, at some fixed strain rate $\dot{\varepsilon}_{p0}$ and temperature T_0. C is a coefficient in the factor for scaling the quasi-static response to higher strain rates.

Thermal softening is accounted for in the final factor. The parameter M controls scaling at temperatures between the reference temperature T_0 and the melt temperature T_m. Above the melt temperature the material is assumed to be a strengthless fluid, for which $\sigma_y = 0$ (note that in ALEGRA this behavior must be explicitly enabled by setting PHASE CONTROL = 1 within the material model). The Johnson–Cook model does not apply at temperatures below the reference temperature, and both codes essentially use max(T, T_0) in place of T in Eq. 12.

CTH employs an additional temperature-related parameter TMELT in the constitutive model. If the material temperature in a cell exceeds TMELT, the constitutive model is bypassed and the flow stress is set to zero. Unfortunately, this parameter was overlooked in the baseline configuration input file. The default value for TMELT is 1490 K. The net effect is that flow stress in the CTH version of the baseline configuration is computed according to the Johnson–Cook model up to 1490 K and is zero for temperatures greater than this. Since this study is only concerned with how simulation outcomes vary with changes in computational factors, this unintended behavior does not impact the validity of this work.

The final aspect of material physics required in these simulations is failure, which can be defined as the loss of ability to sustain shear or tensile loads. Many different models attempt to capture various aspects of failure, such as the accumulation of damage or the onset of brittle fracture due to excessive loading. In this work, simple failure models based on a maximum sustainable tensile pressure were employed.

In CTH, a fracture algorithm in which any cell having tensile pressure exceeding a fixed threshold value is considered failed was used. In ALEGRA, the VOID INSERTION model was employed to similar effect. In addition to a threshold tensile pressure, this model also allows material to fail via dilation below a minimum density limit. This was enabled by setting FORCE FRACTURE = 1, with the density floor set to 80% of the ambient density. The VOID INSERTION model allows the user to specify a relaxation time over which tensile pressure goes to zero after failure. The default behavior of relaxing pressure over 10 cycles was used, though it should be noted that a more realistic approach would be to specify an actual relaxation time since the amount of simulation time spanned by a compute cycle could vary considerably.

Material model parameter values used in this work are summarized in Table 1. EOS model parameters for WHA were based on those for elemental tungsten. Hugoniot fit parameters[16] were modified to account for inclusion of material strength effects in model calculations of the Hugoniot.[17] Note that the ambient density of 19.235 g/cm^3 for tungsten[18] is somewhat greater than that for 93% WHA, which is nominally 17.6 g/cm^3. The value of 0.138 J/gK used for the specific heat capacity is similar to the value of 0.132 J/gK reported for tungsten in the recent reference literature.[19] The Grüneisen parameter can be calculated from Eq. 4 using values for the thermal properties of tungsten found in the reference literature.[19,20] For an isothermal compressibility of 3.09e-12 Pa^{-1}, and taking the volumetric thermal expansion coefficient to be 3 times the linear expansion coefficient of 1.35e-5 K^{-1}, the resulting value for the Grüneisen parameter is 1.7207.

Table 1 Material model parameters for WHA and RHA steel

Parameter	Symbol	WHA	RHA	Units
Initial density	ρ_0	19.235	7.850	g/cm^3
Initial temperature	T_0	298	298	K
Specific heat capacity	C_v	0.138	0.446	J/gK
Grüneisen parameter	Γ_0	1.72	1.67	. . .
Sound speed	Cs	3980	4529	m/s
Mie–Grüneisen model parameters	S_1	1.24	1.49	. . .
	S_2	0	0.00	. . .
Johnson–Cook viscoplastic model parameters	A	1507.0	780.0	MPa
	B	176.6	780.0	MPa
	C	0.016	0.004	. . .
	m	1.00	1.00	. . .
	n	0.120	0.106	. . .
Melt temperature	T_m	1723	1783	K
Poisson's ratio	ν	0.310	0.294	. . .
Fracture strength	P_f	–6.756	–2.500	GPa

Hugoniot fit parameters for 304 stainless steel[21] were used for RHA, with the sound speed decreased slightly to match the value for steel in the Sandia seslan EOS data base.[22] The Grüneisen parameter for 4340 steel was used.[21] The value of 0.446 J/gK used for the specific heat capacity is similar to the value of 0.449 J/gK reported for iron in recent reference literature.[19] The ambient density of 7.85 g/cm^3 used here is nearly the same as the measured average density of 7.84 g/cm^3 for RHA samples reported by Kerley.[23]

Constitutive model parameters, melt temperature, and fracture strength for WHA were taken from Johnson and Holmquist's characterization of 90% WHA material.[24] Note that the melt temperature for WHA is the temperature at which the nickel–iron matrix melts and the alloy loses its strength. Poisson's ratio ν can be estimated by treating WHA as an isotropic, linear elastic material and using the relation

$$\nu = \frac{3K-2G}{6K+2G}\,, \tag{13}$$

where K is the bulk modulus and G is the shear modulus. Johnson and Holmquist provide a shear modulus value of 124 GPa in their model for 90% WHA. The bulk modulus can be derived from the EOS Eq. 11 via the identity

$$K \equiv \rho\frac{dP}{d\rho}\,. \tag{14}$$

This yields the familiar Newton–Laplace equation when evaluated at ambient density and temperature:

$$K = \rho_0 C_s^2 \ . \tag{15}$$

The EOS parameters in Table 1 give a bulk modulus of 304 GPa. Solving Eq. 13 yields a Poisson's ratio of 0.320. A slightly smaller value of 0.310 was used in this work.

Constitutive model parameters, melt temperature, and fracture strength for RHA were taken from Meyer and Kleponis.[25] A Poisson's ratio of 0.294 for steel goes back to Kirchhoff's 19th-century measurements,[26] but is also consistent with modern measurements for a variety of steels, such as 316 stainless.[27]

In all simulations in this work the axis of the threat was aligned with its initial velocity vector. In ballistics, this is commonly referred to as a 0° yaw condition. Target blocks were always aligned normal to and centered on the threat axis, meaning the axis passed through the center point of both the front and back faces of the target. This is known as a 0° obliquity, center-hit condition.

In the baseline configuration of the system, the sides of the target block are aligned to the coordinate axes, with the striking face coincident with the z = 0 plane and centered on the coordinate origin. The projectile is oriented along the z-axis and has a velocity of 1280 m/s in the positive z-direction. The tip of the projectile is initially set back 0.5 cm from the target face.

Domain size in the baseline configuration was set to provide a minimum 2-cm void buffer around the projectile and target. All boundaries are free void, with no symmetry planes used. Simulations of the baseline configuration were run to 300 μs to provide ample time for the penetration process to complete.

Example simulation output for the baseline configuration of the model system is shown in Fig. 3. Images were generated during simulation run time in both CTH and ALEGRA using the integrated Spymaster utility.[28] The state of materials in the system is visualized by coloring domain cells that are at least half-filled with material. RHA is colored orange and WHA is white. All other cells are transparent. Images on the left show a perspective view of the system in 50-μs increments of simulated time. On the right is a cutaway view sectioned along the x = 0 plane to reveal the penetration process within the interior of the block. The positive z-axis is oriented to the right and the positive y-axis is up in these views. Purple dots represent the locations of tracer particles, which are discussed in the next section.

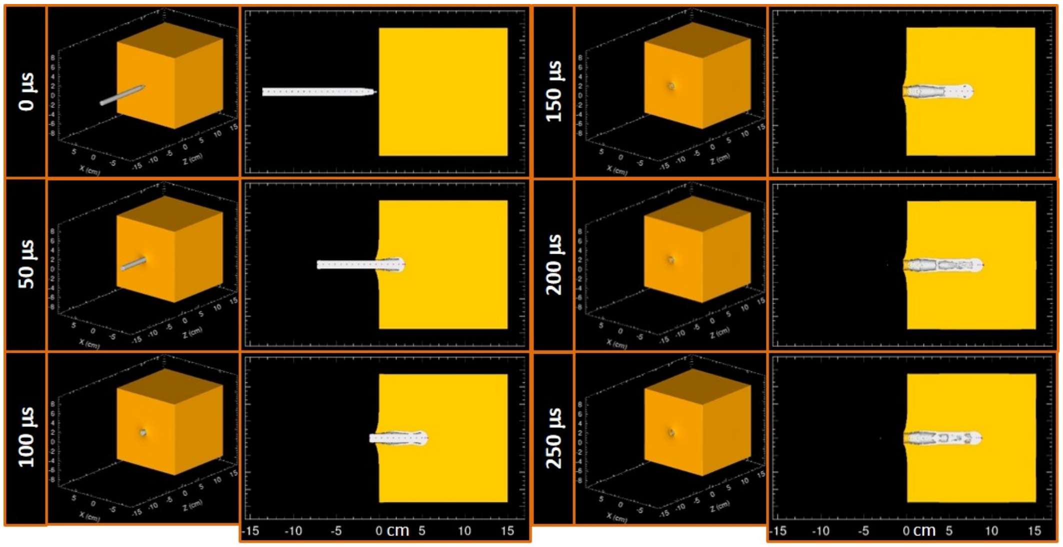

Fig. 3 Perspective (left) and cutaway (right) views in 50-µs increments for a CTH simulation of the model ballistic system in the baseline configuration

A partial CTH input file for the baseline configuration of the system is included in Appendix A. The ALEGRA version is in Appendix B. Table 2 summarizes the simulation parameters for the baseline configuration.

Table 2 Model ballistic system baseline configuration

Resolution (cm)	Void buffer (cm)	Threat offset (cm)	Threat axis	Domain cells				Stop time (µs)
				X	Y	Z	Total	
0.0500	2.00	0.50	+z	380	380	652	94,148,800	300

4. Quantities of Interest

A primary outcome in ballistic simulations is the depth of penetration (DoP) of a threat into a target. Though comparisons with real-world outcomes are not germane to this investigation, the model ballistic system is well characterized experimentally,[10] and the expected DoP for the baseline configuration is 87.9 mm.

DoP can be calculated by measuring the distance from the farthest point of threat advancement to the back-face plane of the target and then subtracting from the initial target thickness. Three different methods for performing this measurement were employed in this work.

The most precise and accurate method directly interrogates domain cell data. In CTH, cell data were extracted using Spymaster's MPISPYPLT postprocessing program. Positions for all cells having at most 99.95% void by volume were recorded. In ALEGRA, the open-source ParaView application[29] was used to extract data for cells containing at least 0.05% by volume of either threat or target material.

Threat position is determined by the location of the cell farthest along the threat axis that contains threat material. Determining the position of the back-face plane of the target is more complicated, especially for simulations in which the threat axis is not aligned with one of the principle domain axes. The general approach was to define a "picture-frame" boundary within a plane normal to the threat axis (Fig. 4). This frame excluded both the central region of the target, which often develops a slight bulging deformation, and the edges of the target, which can lose sharpness of definition due to advection. This picture frame is divided into a grid of cells, and the domain searched along the threat axis to determine the furthest extent of target material in each cell of the frame. The back-face position is then the average value over all cells. This was the most complex of the 3 methods and required the most extensive postprocessing analysis. In all calculations, a 1-cm-wide picture frame was used, offset 0.5 cm from target edges, with cell size equal to twice the resolution of the simulation being analyzed.

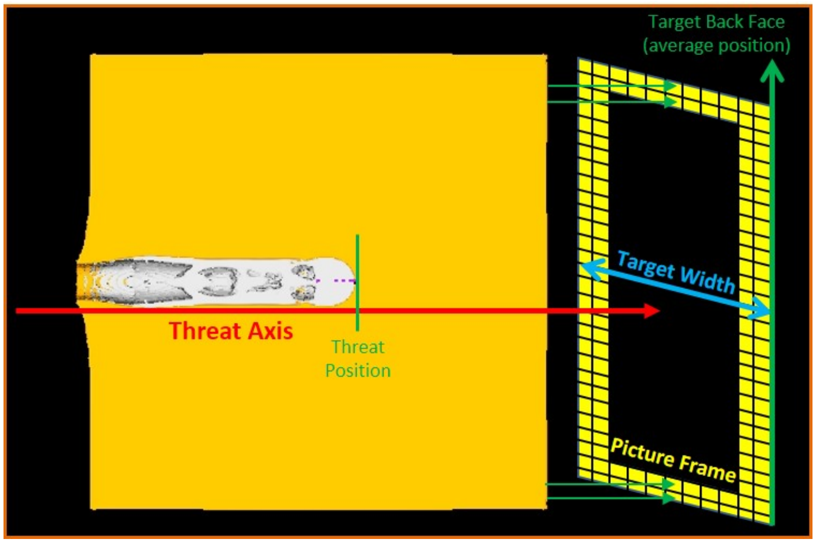

Fig. 4 Illustration of the cell data method for determining DoP

The second method uses tracer particles embedded within the problem. Tracers are massless virtual objects that move in concert with surrounding material and serve as localized point probes. In all simulations, 20 tracers were equally spaced along axis of the threat from tip to tail. For simulations in which the threat was oriented along a principle domain axis, every other tracer was constrained to move only along that axis. Tracer position was written out every 1 µs of simulation time. Threat position at the end of a simulation is the maximum position of all threat tracers (Fig. 5). In addition, tracers were placed at each of the corners of the rear face of the target. These tracers were offset one cell length from each target face to ensure they were fully embedded within the target. The position of the back face at the end of a simulation was computed as the average position along the threat axis of these 4 tracers, adjusted to account for the offset.

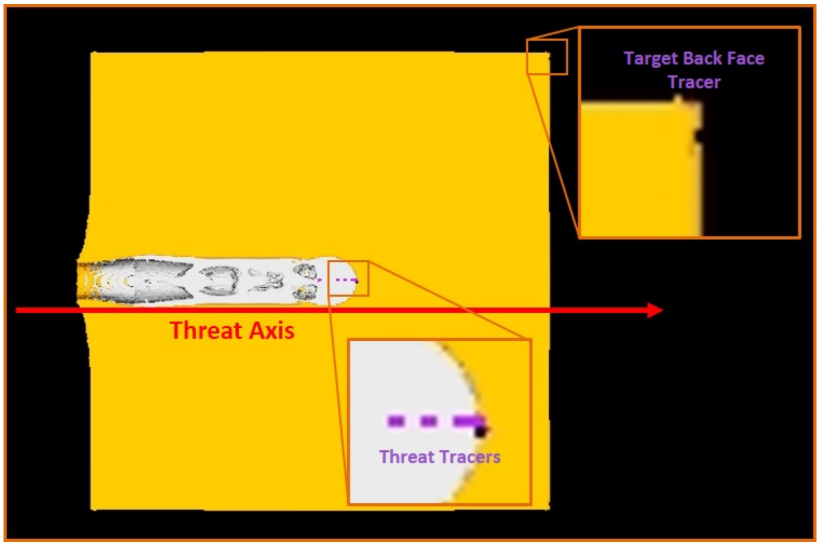

Fig. 5 Illustration of the tracer method for determining DoP

The final method involves examining images of the target sectioned in 1-mm increments along the threat axis to locate both the deepest point of threat advancement and the back face of the target. Postsimulation images in CTH were generated using the Spymaster MPISPYPLT postprocessing program. One-millimeter-thick sections of domain were rendered in 1-mm increments along threat axis from a viewpoint located behind the target (Fig. 6). Materials for threat and target were colored, with all other cells transparent. Slices were sequentially numbered from back to front of the target. The images were inspected to find the first image showing the back face of the target and the first image showing the penetration channel. DoP is then calculated by subtracting the difference of the image indices (–1 to account for viewpoint; see Fig. 7) from the target thickness in millimeters. Since positions are only determined to the nearest millimeter, this method carries a measurement uncertainty of ±1 mm.

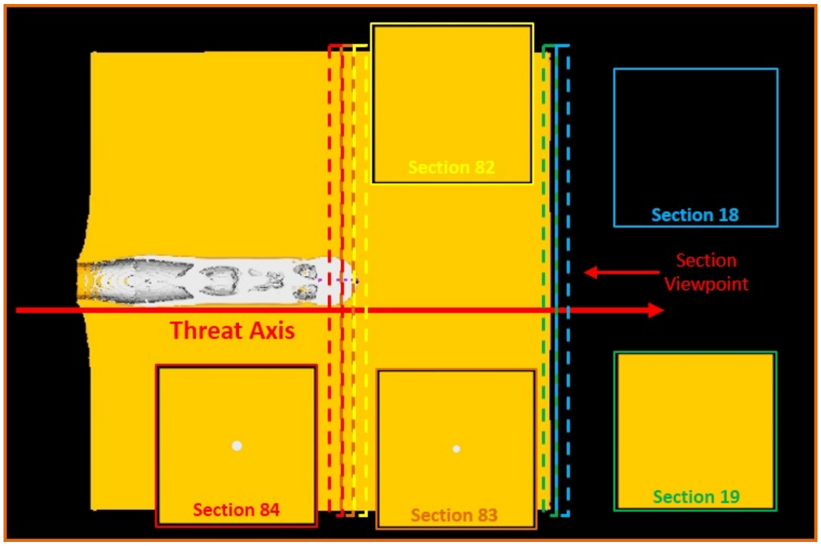

Fig. 6 Illustration of the image method for determining DoP in CTH

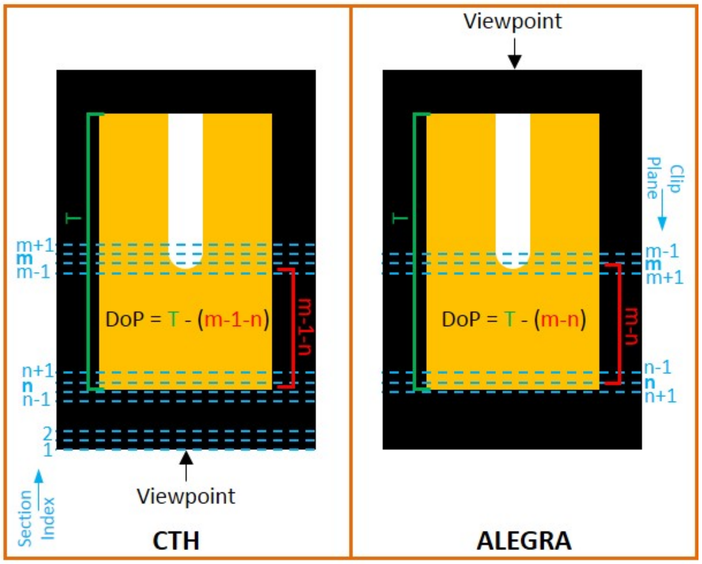

Fig. 7 Illustration of DoP calculation for image method in CTH and ALEGRA

Implementation of this method in ALEGRA differs slightly. Postsimulation images were rendered in ParaView using a series of clipping planes in 1-mm increments along the threat axis, with a viewpoint from the front face of the target. Images were labeled with the position of clip plane. One set of images was created by mapping the density of the threat material and a second set by mapping the density of target material. Images were searched to find the last image containing threat material and the last image containing the target. DoP is then calculated by subtracting the difference of the clip-plane positions from the target thickness (Fig. 7).

DoP is the primary quantity of interest in this work, and the sensitivity of this metric to variations in fundamental computational parameters is the main focus. Several secondary quantities relating to computational performance are also tracked, as such variations can also impact the time required to compute a solution.

Run time refers to the real-world time required to compute a solution. The total computational time for a simulation is run time multiplied by the total number of cores used. Computational time would be a fixed attribute of a simulation in an ideal parallel computer. For example, doubling the number of cores used would reduce the run time by half. In reality, communication of information between cores leads to nonideal scaling. Another measure of the computational cost of a simulation is the number of cycles, or time steps, needed to compute a solution.

Simulations of the baseline configuration for the model system were performed on Hercules. The CTH version was run using 64 cores, while the ALEGRA version was run on 512 cores. Measurements of DoP using the tracer and image methods are consistent with the most accurate values obtained via the cell data method (Table 3). The CTH simulation produces about 4% greater DoP than ALEGRA. The CTH version of the system is computationally smaller by about a factor of 10, using 1142 core-hours compared to 11,981 core-hours for ALEGRA.

Table 3 Simulation results for baseline configuration

Code	Cores				Cycles	Run time (h)	DoP (cm)		
	X	Y	Z	Total			Cell data	Tracers	Images
CTH	4	4	4	64	7771	17.84	8.700	8.69	8.7
ALEGRA	8	8	8	512	8820	23.40	8.450	8.48	8.5

These results form the point of comparison for variations on the baseline configuration and will be listed and highlighted in subsequent tables of results. As an additional note, once the omission of TMELT from the CTH version was discovered, the baseline simulation was repeated with TMELT = 1e30 for both materials so that the constitutive model would apply for all temperatures. This simulation required 7861 cycles over 18.13 h to complete. The resulting DoP values were 8.650 cm for the cell data method, 8.67 cm for the tracer method, and 8.7 cm for the image method.

5. Uncertainty from Variations in Computational Factors

This report focuses on the effects that variations in fundamental computational parameters, many of which tend to be overlooked in routine work, have on simulation outcomes. Such parameters include those affecting the basic structure and execution of simulations. Factors that typically receive high scrutiny, such as material model parameters, are held fixed in this study. Effects due to variances associated with experiments, such as uncertainty in projectile velocity or yaw, are not explored here either. Indeed, one motivation for a detailed study of basic computational factors is to isolate their influence in preparation for future investigations into these other sources of uncertainty.

Results in this section are subdivided into 4 general categories. Computational Execution covers factors involved in running a simulation that are external to the input file, such as the choice of platform and number of cores used. Computational Domain includes parameters affecting the overall structure of the domain and the spatial placement of simulated objects, such as resolution and the location of the coordinate origin. Time-Step Control groups factors affecting the temporal progress of a simulation. Finally, Physical Invariance covers tests of code fidelity to outcome invariances in the real world, such as those related to changing frames of reference.

5.1 Computational Execution

On all platforms used in this study, simulations are submitted to an automated scheduling system that queues requests and manages the computing resources of the platform. The submissions set the number of cores used for a simulation and how many cores are active on each compute node. Only parameters relating to execution of simulations are varied in this subsection. In all cases, the same baseline configuration input file was used without alteration.

The first study performed was the most basic. A presumption of any computing hardware is uniformity over time. On Hercules, that means compute node number 1 functions identically to compute node number 1092 and the same from the day the machine is commissioned to the day it is decommissioned.

Multiple simulations of the baseline configuration were submitted and run concurrently as a test of this presumption. Furthermore, repeat simulations of the baseline configuration were periodically performed throughout the course of this study. Table 4 shows the results of this replication study for CTH.

Table 4 CTH replication study results

Run	Cycles	Run time (h)	DoP (cm)		
			Cell data	Tracers	Images
BL	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
2	7771	17.85	8.700	8.69	8.7
3	7771	17.87	8.700	8.69	8.7
4	7771	17.84	8.700	8.69	8.7
5	7771	17.83	8.700	8.69	8.7
6	7771	17.87	8.700	8.69	8.7
7	7771	17.82	8.700	8.69	8.7
8	7771	17.82	8.700	8.69	8.7
9	7771	17.95	8.700	8.69	8.7
10	7771	17.84	8.700	8.69	8.7
11	7771	17.94	8.700	8.69	8.7
12	7771	17.92	8.700	8.69	8.7
13	7771	17.89	8.700	8.69	8.7

Run BL, highlighted in bold, is the baseline configuration simulation from Table 3. Runs 2 and 3 were replications that ran concurrently, meaning they necessarily used different nodes of the platform. Similarly, runs 4–6 ran concurrently. Runs 7–13 were replicates performed at various later times, with all simulations performed over a period of time spanning 13 January 2016 to 8 June 2016.

All simulations required 7771 cycles to complete and generated identical DoP results, indicating perfect replication. A close examination of the CTH output files bolsters this claim significantly. A standard CTH run-time output is a running log that records for each cycle the cycle number, simulation time (in scientific notation with 12 significant digits), and time step (same format). This output for cycle 7771 was identical for all runs, which is virtually impossible unless each simulation exactly reproduced the billions of calculations required to reach completion.

The average run time for these 13 simulations was 17.87 h, with a standard deviation of just 2.67 min, which is 0.25% of the mean. This provides a measure of the inherent variance in simulation run time.

Results of the replication study for ALEGRA are shown in Table 5. The baseline and runs 2–6 all ran concurrently. All simulations were performed from 13 July 2016 to 25 October 2016. Like CTH, ALEGRA also logs information during run time, and recorded values for cycle 8820 were identical for all simulations.

Table 5 ALEGRA replication study results

Run	Cycles	Run time (h)	DoP (cm)		
			Cell data	Tracers	Images
BL	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
2	8820	23.40	8.450	8.48	8.5
3	8820	23.41	8.450	8.48	8.5
4	8820	23.30	8.450	8.48	8.5
5	8820	23.54	8.450	8.48	8.5
6	8820	23.61	8.450	8.48	8.5
7	8820	26.08	8.450	8.48	8.5
8	8820	23.43	8.450	8.48	8.5
9	8820	23.69	8.450	8.48	8.5
10	8820	23.32	8.450	8.48	8.5
11	8820	23.39	8.450	8.48	8.5
12	8820	23.37	8.450	8.48	8.5

In terms of run time, run 7 is a clear outlier. Delays between the execution of a submission by the scheduler, when the start time used in this work is recorded, and the commencement of the actual simulation can occur for a number of reasons, though the specific cause in this case is unknown. The average run time excluding run 7 was 23.44 h, with a standard deviation of 7.27 min, which is 0.52% of the mean.

Next, the baseline simulation was repeated on different platforms. The CTH baseline simulation was repeated 3 times on Pershing (Table 6). Solutions were identical to those on Hercules, though run times on Pershing were a few hours longer. The simulation was run twice on Excalibur. While all CTH simulations used 64 total cores over 4 nodes, Excalibur has twice as many cores on each compute node as Hercules and Pershing. One simulation was performed using all 32 cores on each of 2 nodes, and a second was done using just 16 cores per node over 4 nodes. The differing number of cycles demonstrates that Excalibur produces a different solution than Hercules and Pershing. However, the only measureable effect on DoP is a negligible change in final tracer position. Results for ALEGRA simulations on different platforms (Table 7) behave in the same manner as those for CTH.

Table 6 CTH platform study results

Run	Platform	Cores/ node	Cycles	Run time (h)	DoP (cm)		
					Cell data	Tracers	Images
BL	**Hercules**	**16**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
2	Pershing	16	7771	19.58	8.700	8.69	8.7
3	Pershing	16	7771	19.66	8.700	8.69	8.7
4	Pershing	16	7771	19.69	8.700	8.69	8.7
5	Excalibur	32	7778	18.11	8.700	8.70	8.7
6	Excalibur	16	7778	17.92	8.700	8.70	8.7

Table 7 ALEGRA platform study results

Run	Platform	Cores/ node	Cycles	Run time (h)	DoP (cm)		
					Cell data	Tracers	Images
BL	**Hercules**	**16**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
2	Pershing	16	8820	25.07	8.450	8.48	8.5
3	Pershing	16	8820	25.03	8.450	8.48	8.5
4	Pershing	16	8820	25.03	8.450	8.48	8.5
5	Excalibur	32	8813	23.50	8.450	8.47	8.5
6	Excalibur	16	8813	23.33	8.450	8.47	8.5

Following on the decision to repeat the Excalibur simulation using different numbers of cores per node, a closer look at this parameter was undertaken on Hercules. Reducing the number of cores per node has 2 effects. The first is to divide the fixed amount of memory on each node among fewer cores. The second effect is to increase the number of nodes used in the computation.

Results for CTH (Table 8) and ALEGRA (Table 9) demonstrate that changing the number of cores/node does not affect the solution state of this model system. While there is a modest improvement in run times as the memory per core increases, the subsequent study demonstrates that it is better by far to utilize the full core capacity of each node used in a simulation.

Table 8 CTH cores-per-node study results

Run	Cores/ node	Total nodes	Cycles	Run time (h)	DoP (cm)		
					Cell data	Tracers	Images
BL	**16**	**4**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
2	8	8	7771	17.50	8.700	8.69	8.7
3	4	16	7771	16.08	8.700	8.69	8.7
4	2	32	7771	15.29	8.700	8.69	8.7

Table 9 ALEGRA cores-per-node study results

Run	Cores/ node	Total nodes	Cycles	Run time (h)	DoP (cm)		
					Cell data	Tracers	Images
BL	**16**	**32**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
2	8	64	8820	23.11	8.450	8.48	8.5
3	4	128	8820	20.30	8.450	8.48	8.5

The most significant factor in computational execution is the total number of cores used for a simulation. The idea underlying parallel computing is that instead of a single core grinding through all the calculations necessary to advance a simulation from one time step to the next, many cores can simultaneously calculate smaller subsets of the problem in a shorter amount of time. In principle, a calculation that takes a single core 10 h to perform can be accomplished by 10 cores in just 1 h. In practice, the scaling is less than ideal due to the need to ensure boundary values across adjoining pieces of the computational domain are shared. This requires communication of information between cores, which adds additional time to the computation.

The simulation codes attempt to partition the computational domain such that each core is assigned approximately the same number of cells, while minimizing the number of boundary cells that need to be communicated between cores. In box-shaped domains composed of uniform cells, such as those used exclusively in this work, decomposition takes the form of a grid along each coordinate axis. An illustration showing a 2 × 3 × 6 domain decomposition (for 36 total cores) is provided in Fig. 8.

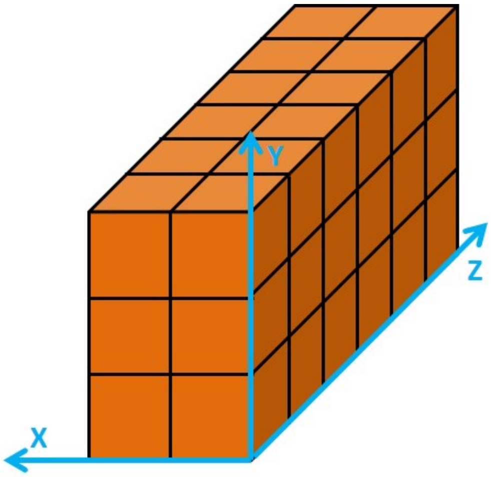

Fig. 8 Illustration of domain decomposition

Table 10 shows results of the baseline simulation in CTH across a range of total cores from 8 to 2048. The associated decompositions are the default generated by the code. Both codes allow the user to force a specified decomposition, but this requires a change to the input files, and so investigation of that capability is presented in the next section.

Table 10 CTH total cores study results

Run	Cores	Cores/ node	Decomposition			Cycles	Run time (h)	DoP (cm)		
			X	Y	Z			Cell data	Tracers	Images
2	8	8	2	2	2	7812	97.13	8.700	8.70	8.7
3	9	9	1	3	3	7810	85.20	8.700	8.69	8.7
4	10	10	1	2	5	7811	76.89	8.700	8.69	8.7
5	11	11	1	1	11	7812	76.87	8.700	8.69	8.7
6	12	12	2	2	3	7810	58.93	8.700	8.70	8.7
7	13	13	1	1	13	7812	66.96	8.700	8.69	8.7
8	14	14	1	2	7	7809	54.89	8.700	8.69	8.7
9	15	15	1	3	5	7810	56.92	8.700	8.69	8.7
10	16	16	2	2	4	7816	50.13	8.700	8.69	8.7
11	31	1	1	1	31	7812	27.90	8.700	8.69	8.7
12	32	16	2	4	4	7774	29.15	8.700	8.70	8.7
BL	**64**	**16**	**4**	**4**	**4**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
13	128	16	4	4	8	7775	10.07	8.700	8.69	8.7
14	256	16	4	8	8	7812	5.50	8.700	8.69	8.7
15	512	16	8	8	8	7809	3.21	8.700	8.69	8.7
16	1024	16	8	8	16	7811	2.98	8.700	8.70	8.7
17	2048	16	8	16	16	7810	1.87	8.700	8.70	8.7

It is clear from the results that solutions in CTH vary with total number of cores. For this model system, the differences are not enough to affect the DoP measurements, apart from some negligible jitter in final tracer position. Simulation run time generally scales with total cores (Fig. 9), but the scaling is not monotonic for core counts below 32. From 32 to 512 cores, doubling the core count roughly halves the run time. Above 512 cores, improvements in run time are inefficient, as demonstrated by the large increases in total computational time of the simulation.

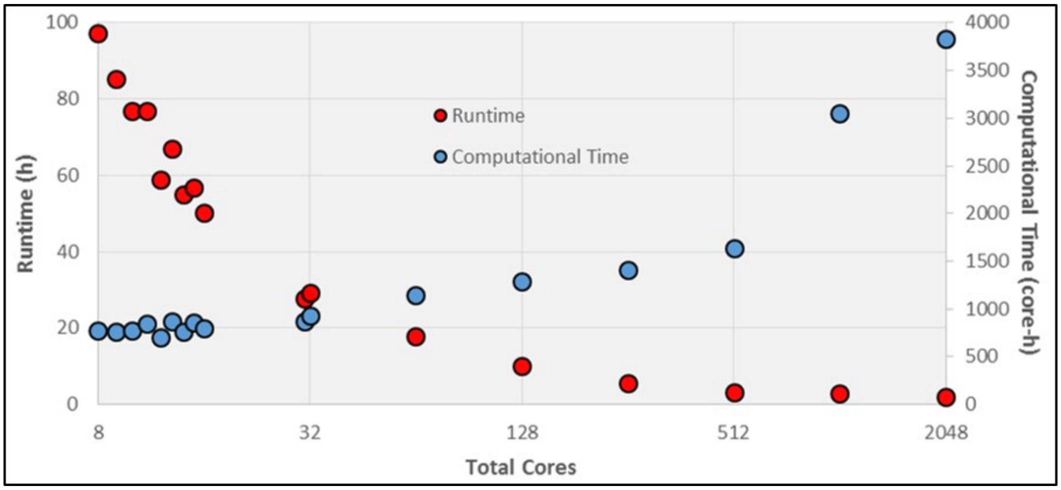

Fig. 9 Plot of run time and computational time vs. total cores (plotted logarithmically) for the CTH total core study

Results for ALEGRA are shown in Table 11. The cores per node had to be reduced to 8 to provide enough memory per core for the simulation to run on only 128 cores. Examination of the output data logs shows that all solutions are mathematically identical, in contrast with CTH behavior. Near ideal scaling in run time held across the range of total cores examined.

Table 11 ALEGRA total cores study results

Run	Cores	Cores/ node	Decomposition			Cycles	Run time (h)	DoP cm)		
			X	Y	Z			Cell data	Tracers	Images
2	128	8	4	4	8	8820	88.28	8.450	8.48	8.5
3	256	16	4	8	8	8820	44.91	8.450	8.48	8.5
BL	**512**	**16**	**8**	**8**	**8**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
4	1024	16	8	8	16	8820	12.32	8.450	8.48	8.5
5	2048	16	8	16	16	8820	6.60	8.450	8.48	8.5

5.2 Computational Domain

To isolate the effects of decomposition from those related to changing the total number of cores, simulations having different user-specified decompositions were performed using a fixed number of total cores.

CTH results (Table 12) demonstrate that how the simulation is divvied up among cores can significantly impact run time. A general trend for this model system is that run times favor decomposition along the *z*-axis, which is the threat axis in the base configuration, as well as the longest dimension of the computational domain. Note that the default CTH decomposition does not have the shortest run time. Thus, for problems requiring large numbers of simulations, it may be advantageous for the user to try different decompositions at the outset to reduce the computational time.

Table 12 CTH decomposition study results

Run	Decomposition			Cycles	Run time (h)	DoP (cm)		
	X	Y	Z			Cell data	Tracers	Images
2	1	1	64	7812	22.61	8.700	8.69	8.7
3	1	64	1	7816	27.80	8.700	8.70	8.7
4	64	1	1	7812	34.60	8.700	8.69	8.7
5	2	2	16	7810	15.07	8.700	8.70	8.7
6	2	16	2	7813	16.91	8.700	8.69	8.7
7	16	2	2	7811	20.91	8.700	8.69	8.7
8	1	8	8	7811	16.46	8.700	8.69	8.7
9	8	1	8	7815	19.13	8.700	8.69	8.7
10	8	8	1	7810	15.96	8.700	8.70	8.7
BL	**4**	**4**	**4**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**

The ALEGRA simulations, unlike CTH, produce mathematically identical solutions under different decompositions (Table 13). The default decomposition also clocks in with the shortest run time among the examined alternatives.

Table 13 ALEGRA decomposition study results

Run	Decomposition			Cycles	Run time (h)	DoP (cm)		
	X	Y	Z			Cell data	Tracers	Images
2	2	2	128	8820	28.19	8.450	8.48	8.5
3	2	128	2	8820	37.60	8.450	8.48	8.5
4	128	2	2	8820	40.94	8.450	8.48	8.5
5	4	4	32	8820	23.91	8.450	8.48	8.5
6	4	32	4	8820	25.57	8.450	8.48	8.5
7	32	4	4	8820	26.51	8.450	8.48	8.5
BL	**8**	**8**	**8**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**

The model ballistic system was constructed to be entirely embedded within the computational domain so that materials did not encounter the domain boundaries. This was a deliberate choice, as boundary cells are treated differently than interior ones. Boundary cells are subject to user-selected conditions to compensate for having less than a full complement of neighboring cells. Maintaining a void buffer limits the influence of these conditions on the simulation solution.

Adding a void buffer also increases the size of the domain. To examine the computational cost of this, simulations with buffers of different size were performed. In both CTH (Table 14) and ALEGRA (Table 15), the amount of void buffer changes the solution state but has no measureable effect on DoP.

Table 14 CTH void buffer study results

Run	Void buffer (cm)	Domain cells				Cycles	Run time (h)	DoP (cm)		
		X	Y	Z	Total (M)			Cell data	Tracers	Images
2	0.50	320	320	592	60.6	7782	12.02	8.700	8.69	8.7
BL	**2.00**	**380**	**380**	**652**	**94.1**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
3	8.00	620	620	892	342.9	7788	58.35	8.700	8.69	8.7

Table 15 ALEGRA void buffer study results

Run	Void buffer (cm)	Domain cells				Cycles	Run time (h)	DoP (cm)		
		X	Y	Z	Total (M)			Cell data	Tracers	Images
2	0.50	320	320	592	60.6	8815	15.63	8.450	8.48	8.5
BL	**2.00**	**380**	**380**	**652**	**94.1**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
3	8.00	620	620	892	342.9	8816	41.13	8.450	8.48	8.5

Run time does increase with buffer size. However, differences in scaling between the codes are evident. In both CTH and ALEGRA, increasing the buffer from 0.5 to 2.0 cm increases the run time by about 50%, in proportion to the increase in total number of cells. This proportionality continues to hold in CTH going from 2.0 to 8.0 cm as the run time increases by an additional factor of more than 3, while the run time in ALEGRA increases by only a factor of about 1.75.

The computational size of a simulation possessing mirror plane symmetry can be significantly reduced. This is done by truncating the computational domain at the symmetry plane and imposing a mirror boundary condition. In the baseline configuration of the model ballistic system, both the $x = 0$ and $y = 0$ planes are mirror symmetry planes. Each can be used to reduce the computational time by a factor of 2.

Simulations employing *x*-symmetry, *y*-symmetry, and both simultaneously were performed (Tables 16 and 17). The total number of cores was scaled along with symmetry to keep the partitioning of the problem fixed.

Table 16 CTH symmetry study results

Run	Symmetry	Domain cells			Cores	Cycle s	Run time (h)	DoP (cm)		
		X	Y	Z				Cell data	Tracers	Images
BL	**None**	**380**	**380**	**652**	**64**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
2	X	190	380	652	32	7781	16.95	8.700	8.70	8.7
3	Y	380	190	652	32	7771	17.05	8.700	8.70	8.7
4	X&Y	190	190	652	16	7760	17.00	8.700	8.69	8.7

Table 17 ALEGRA symmetry study results

Run	Symmetry	Domain cells			Cores	Cycles	Run Time (h)	DoP (cm)		
		X	Y	Z				Cell data	Tracers	Images
BL	**None**	**380**	**380**	**652**	**512**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
2	X	190	380	652	256	8798	22.53	8.450	8.47	8.5
3	Y	380	190	652	256	8798	22.54	8.450	8.47	8.5
4	X&Y	190	190	652	128	8755	22.56	8.450	8.47	8.5

Exploiting symmetry planes changes the solution states, but has negligible effect on DoP values in the model system. There is a small improvement in run time beyond that due to reduction of domain size. This is consistent with the fact that reducing the number of total cores also reduces the total computational time of a simulation, as shown by the data plotted in Fig. 9.

The orientation of the coordinate system in the simulation codes is tied to the domain structure, with coordinate axes aligned with cell edges. However, the position of the coordinate origin is set by the user. Since object geometries are defined relative to the origin, a shift in the origin point will translate objects within the domain.

The origin position is fixed by specifying the minimum coordinate values in the domain. In the baseline configuration, the minimum coordinate values were $x_{\mathrm{min}} = y_{\mathrm{min}} = –9.50$ cm and $z_{\mathrm{min}} = –15.60$ cm. This placed the origin between cells 190 and 191 in *x* and *y*, and between cells 312 and 313 along *z*. This location corresponds to a corner of a cell, which is a node point of the mesh.

Simulations were run to examine the effects of shifting the origin small distances in different directions away from this node (Tables 18 and 19). "Shift Type" denotes the displacement of the origin from the baseline nodal position, with "dx" denoting one cell length in the *x*-direction, "dy" one cell length in the *y*-direction, and so on. The origin was placed at different cell symmetry points, such as corners and face centers, as well as at an arbitrary interior point.

Table 18 CTH origin shift study results

Run	Shift type	Cycles	Run time (h)	DoP (cm)		
				Cell data	Tracers	Images
BL	**None**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
2	-dx	7785	17.85	8.700	8.69	8.7
3	-dy	7813	18.01	8.700	8.69	8.7
4	-dz	7783	17.90	8.700	8.69	8.7
5	-dx/2	8132	18.77	8.700	8.69	8.7
6	-dy/2	8350	19.23	8.700	8.69	8.7
7	-dz/2	7756	17.81	8.700	8.70	8.7
8	-(dx+dy)/2	8514	19.63	8.700	8.68	8.7
9	-(dx+dz)/2	8033	18.43	8.700	8.70	8.7
10	-(dy+dz)/2	7759	17.83	8.700	8.70	8.7
11	-(dx+dy+dz)/2	8237	18.90	8.700	8.70	8.7
12	-(dx+dy+dz)/4	7982	18.34	8.699	8.69	8.7
13	-0.176dx-0.57dy-0.6dz	7915	18.24	8.700	8.71	8.8

Table 19 ALEGRA origin shift study results

Run	Shift type	Cycles	Run time (h)	DoP (cm)		
				Cell data	Tracers	Images
BL	**None**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
2	-dx	8807	23.88	8.450	8.47	8.5
3	-dy	8801	23.51	8.450	8.47	8.5
4	-dz	8813	24.17	8.450	8.48	8.5
5	-dx/2	9138	24.64	8.500	8.49	8.5
6	-dy/2	9183	24.86	8.500	8.49	8.5
7	-dz/2	8826	23.79	8.500	8.48	8.5
8	-(dx+dy)/2	9085	24.27	8.400	8.43	8.4
9	-(dx+dz)/2	9287	24.94	8.500	8.49	8.5
10	-(dy+dz)/2	9271	24.83	8.500	8.49	8.5
11	-(dx+dy+dz)/2	8904	23.58	8.450	8.44	8.4
12	-(dx+dy+dz)/4	9278	24.94	8.450	8.49	8.4
13	-0.176dx-0.57dy-0.6dz	10131	27.28	8.500	8.48	8.5

The CTH results exhibit a 1.8-h spread in run times, which is 10% of the mean value of 18.4 h, while the ALEGRA results have a 3.9-h spread, 16% of the 24.5-h mean run time. This spread in run time is caused mainly by the variance in the number of cycles needed to reach completion. The average computational rates remain largely unaffected: 433–436 cycles/h for the CTH simulations and 365–378 cycles/h for the ALEGRA simulations. Origin placement clearly has an effect on run time, though no definitive pattern linking specific origin positions to resulting run time is evident.

The ALEGRA simulations exhibit the first notable changes in DoP results computed with the cell data method. Some part of this may be due to the translation of objects with the coordinate origin. A subresolution origin shift can drag object boundaries across cell boundaries, which registers as a 1-cell change in position within the discretized grid of the domain.

While many of the factors investigated in this study are easily overlooked, domain resolution is one of the most often discussed parameters in simulation work. The effects of resolution on simulation outcomes are both sizeable and well studied.

Real physical systems are generally described by sets of equations involving continuous variables, such as time and position. Computational simulations require that these equations be replaced by versions that operate on discrete values. For example, differential equations become difference equations, and integrals become summations. These discretized equations are fundamentally approximations of the original continuum equations and include some amount of error.

A key postulate underlying numerical simulations is that as the size of the discrete elements (the cell size) becomes vanishingly small, the approximation errors also vanish, and solutions for the set of discretized equations converge to the solution for the original continuum equations. This can be expressed mathematically as

$$\lim_{r \to 0} S_r = S_0 \ , \tag{16}$$

where S_r is the solution state for a simulation performed at resolution r, and S_0 is the presumed solution of the original set of continuum equations. This behavior only holds for systems that meet certain mathematical criteria, which includes limits on the size of allowed time steps (as discussed further in Section 5.3). Not all simulations satisfy these criteria, which is one reason why simulations can fail to produce a reasonable solution or any solution at all. However, in well-behaved simulations, the error is bounded as per the limiting behavior in Eq. 16.

In principle the bound on the error can be calculated. In practice this is generally not feasible, as it requires direct analysis of the full set of discretized equations, which can be quite extensive. Furthermore, no number of simulations can actually prove that solutions are “close” to the continuum solution. The best that can typically be accomplished is to demonstrate that at some resolution, differences in simulation outcomes at nearby resolutions are within some acceptable tolerance.

A study was performed in which cell size was decreased from a starting value of 0.1 cm by factors of the cube root of 2 until simulations became too large to run politely on a shared platform. The scaling factor corresponds to a doubling of the total number of cells in the domain at each step. The total number of cores used was scaled with domain size. Results are shown in Tables 20 and 21.

Table 20 CTH resolution study results

Run	Resolution (cm)	Domain cells			Cores	Cycles	Run time (h)	DoP (cm)		
		X/Y	Z	Total(M)				Cell data	Tracers	Images
2	0.1000	190	326	11.8	8	4095	6.91	8.200	8.15	8.2
3	0.0794	240	411	23.7	16	5293	9.49	8.333	8.38	8.5
4	0.0630	302	518	47.2	32	6662	13.00	8.574	8.54	8.6
BL	**0.0500**	**380**	**652**	**94.1**	**64**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
5	0.0397	479	822	188.6	128	10821	25.94	8.770	8.78	8.8
6	0.0315	604	1035	377.6	256	12403	30.59	8.795	8.80	8.8
7	0.0250	760	1304	753.2	512	27914	70.32	8.825	8.81	8.8
8	0.0198	958	1643	1507.9	1024	51686	132.70	8.829	8.83	8.9
9	0.0157	1207	2070	3015.7	4096	104310	163.27	8.827	8.82	8.9

Table 21 ALEGRA resolution study results

Run	Resolution (cm)	Domain cells			Cores	Cycles	Run time (h)	DoP (cm)		
		X/Y	Z	Total(M)				Cell data	Tracers	Images
2	0.1000	190	326	11.8	64	4715	11.39	7.500	7.44	7.4
3	0.0794	240	411	23.7	128	5429	13.63	7.777	7.80	7.7
4	0.0630	302	518	47.2	256	6883	17.80	8.259	8.22	8.2
BL	**0.0500**	**380**	**652**	**94.1**	**512**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
5	0.0397	479	822	188.6	1024	11602	32.15	8.492	8.51	8.5
9	0.0315	604	1035	377.6	2048	15646	45.05	8.606	8.60	8.6

Computational time increases quickly with decreasing cell size (Fig. 10). In addition to the doubling of the number of cells at each increment, smaller cells necessitate smaller time steps, and thus more cycles are needed to reach completion. The increasing number of cores required to run the simulation imposes additional costs, as demonstrated previously in Fig. 9.

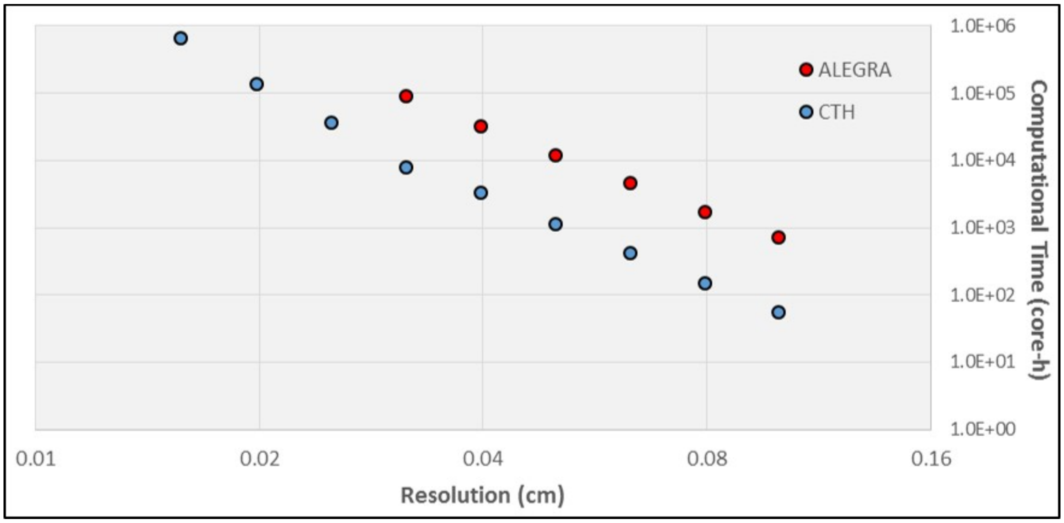

Fig. 10 Log-log plot of computational time vs. resolution

DoP results vary considerably with resolution, with generally increasing penetration as cell size decreases (Fig. 11). At the finest resolutions in CTH, the DoP appears to plateau near 8.83 cm, about 1.5% more than the result for the baseline configuration. In many applications, this difference would not be sufficient to necessitate running simulations at the much costlier resolution. The ALEGRA results do not appear to close on a limiting value over the range of resolutions investigated, but the difference between the ALEGRA results and the CTH results decreases.

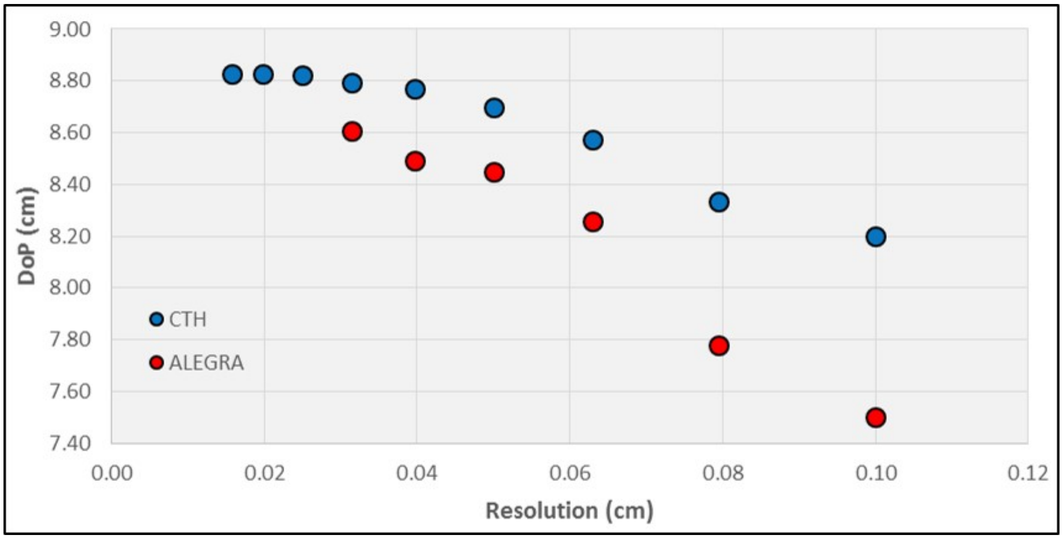

Fig. 11 Plot of DoP vs. resolution

5.3 Time-Step Control

There is some ambiguity regarding when a simulation is considered "complete". In a penetration problem, simulations should be run at least to such time as plastic material flow ceases. However, elastic vibrations can persist long after the penetration process halts. To determine if simulation outcomes are affected by the choice of stop time, simulations were run with stop times ranging from 250 μs to as long as 1000 μs (Tables 22 and 23). In both codes the DoP results are invariant to increasing stop time, apart from a single cell variation seen in the ALEGRA results.

Table 22 CTH stop time study results

Run	**Stop time (μs)**	**Cycles**	**Run time (h)**	**DoP (cm)**		
				Cell data	**Tracers**	**Images**
2	250	6509	15.07	8.700	8.70	8.7
3	275	7140	16.43	8.700	8.70	8.7
BL	**300**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
4	325	8401	19.24	8.700	8.70	8.8
5	350	9031	20.80	8.700	8.70	8.8
6	400	10292	23.64	8.700	8.70	8.7
7	500	12812	29.24	8.700	8.69	8.7
8	750	19107	43.42	8.700	8.70	8.7
9	1000	25400	57.84	8.700	8.70	8.8

Table 23 ALEGRA stop time study results

Run	**Stop time (μs)**	**Cycles**	**Run time (h)**	**DoP (cm)**		
				Cell data	**Tracers**	**Images**
2	250	7374	19.52	8.450	8.48	8.4
3	275	8097	21.47	8.500	8.48	8.5
BL	**300**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
4	325	9544	25.78	8.500	8.49	8.5
5	350	10270	27.13	8.450	8.48	8.4
6	400	11721	31.03	8.500	8.49	8.4
7	500	14608	38.68	8.500	8.47	8.5
8	750	21808	58.39	8.500	8.48	8.5
9	1000	29003	76.63	8.450	8.49	8.3

Time steps in both codes are calculated using stability criteria for discrete calculations. In broad terms, time steps are limited by the need to prevent information from traversing more than a single cell during a cycle. Users can also set a maximum limit on the size of a time step. Simulations were run using a range of maximum time steps to investigate effects on outcomes (Tables 24 and 25).

Table 24 CTH maximum time step study results

Run	Max time step (ns)	Cycles	Run time (h)	DoP (cm)		
				Cell data	Tracers	Images
BL	**None**	**7771**	**17.84**	**8.700**	**8.69**	**8.7**
2	40	7771	17.92	8.700	8.69	8.7
3	35	8592	20.00	8.700	8.69	8.7
4	30	10007	23.03	8.700	8.70	8.7
5	25	12001	27.66	8.700	8.70	8.7
6	20	15001	34.55	8.700	8.70	8.7
7	15	20000	45.86	8.700	8.70	8.7
8	10	30001	68.65	8.700	8.71	8.8

Table 25 ALEGRA maximum time step study results

Run	Max time step (ns)	Cycles	Run time (h)	DoP (cm)		
				Cell data	Tracers	Images
BL	**None**	**8820**	**23.40**	**8.450**	**8.48**	**8.5**
2	45	8820	23.38	8.450	8.48	8.5
3	40	8813	23.33	8.450	8.48	8.5
4	35	8853	23.54	8.450	8.48	8.5
5	30	10083	26.66	8.450	8.48	8.5
6	25	12028	31.95	8.450	8.47	8.5
7	20	15001	39.79	8.450	8.47	8.5
8	15	20000	52.97	8.450	8.47	8.5
9	10	30001	79.18	8.450	8.48	8.5

The maximum time step for run 2 in both codes was set to be larger than the maximum time step encountered in the baseline configuration simulations. As would be expected, these simulations produce solutions that are mathematically identical to those for the baseline simulations. Apart from the unusual instance of ALEGRA run 3, further restricting the maximum time step naturally increases the number of cycles required to reach completion. However, DoP results remain unchanged.

Sometimes simulations fail to run to completion. For especially large or complex simulations, this can result in large amounts of lost computational time. To mitigate this possibility both codes allow periodic recording of the simulation state.

Disrupted calculations can then be restarted from these intermediate states, avoiding the need to repeat the entire simulation from the beginning. Ideally, restarting a simulation from an intermediate state will have no effect on outcomes. To examine the effects of restarting an interrupted simulation, simulations were run as a series of restarts every 50 μs. DoP results were calculated for each increment beginning at 250 μs (Tables 26 and 27).

Table 26 CTH restart study results

Run	Restart	Stop time (μs)	Cycles	Run time (h)	Cumulative	DoP (cm)		
						Cell data	Tracers	Images
2	0	50	1361	2.96	2.96	. . .	. . .	. . .
3	1	100	2672	3.15	6.11	. . .	. . .	. . .
4	2	150	3965	3.07	9.18	. . .	. . .	. . .
5	3	200	5232	2.92	12.10	. . .	. . .	. . .
6	4	250	6509	2.90	15.00	8.700	8.70	8.7
7	5	300	7771	2.84	17.84	8.700	8.69	8.7
BL	**. . .**	**300**	**7771**	**17.84**	**17.84**	**8.700**	**8.69**	**8.7**
8	6	350	9031	2.84	20.68	8.700	8.70	8.8
9	7	400	10292	2.84	23.52	8.700	8.70	8.7

Table 27 ALEGRA restart study results

Run	Restart	Stop time (μs)	Cycles	Run time (h)	Cumulative	DoP (cm)		
						Cell data	Tracers	Images
2	0	50	1425	3.70	3.70	. . .	. . .	. . .
3	1	100	2862	3.84	7.54	. . .	. . .	. . .
4	2	150	4462	4.30	11.84	. . .	. . .	. . .
5	3	200	5907	3.86	15.70	. . .	. . .	. . .
6	4	250	7374	3.92	19.62	8.450	8.48	8.4
7	5	300	8820	3.83	23.45	8.450	8.48	8.5
BL	**. . .**	**300**	**8820**	**23.40**	**23.40**	**8.450**	**8.48**	**8.5**
8	6	350	10270	3.83	27.28	8.450	8.48	8.4
9	7	400	11721	3.84	31.12	8.500	8.49	8.4

Comparison of runs 6–9 with the corresponding simulations from the stop time study (Tables 22 and 23) show that the restart series produces solutions that are mathematically identical to simulations that run without interruption. Restarting a calculation from an intermediate state has no effect on model system solutions.

5.4 Physical Invariance

The final studies in this report examine the fidelity of the simulation codes to fundamental physical laws. According to current scientific understanding of the universe, the physics of objects are invariant to translations and rotations in space, attributes intimately related to momentum conservation laws. Additional physical principles applicable to simulations include invariance of physics to transformations from one inertial reference frame to another (Galilean invariance) and Newton's first law of motion, also known as the law of inertia.

Translations in space were indirectly involved in the origin shift study in Section 5.2. A more direct test of translational invariance was conducted in which the model system geometry was shifted wholesale in various directions within a domain having a fixed coordinate system. To provide sufficient space for translations, an expanded domain composed of 620 × 620 × 892 cells was employed, identical to the largest domain in the void buffer study (run 3 in Tables 14 and 15).

Results for both codes demonstrate invariance under spatial translation (Tables 28 and 29). An interesting quirk in the CTH simulations is a 10% reduction in run time for any translations that include a displacement in the *z*-direction. In contrast, simulations in ALEGRA exhibit little variance in run time, with all simulations completing within 0.75 h of the mean.

Table 28 CTH translation study results

Run	Translation (cm)			Cycles	Run time (h)	DoP (cm)		
	X	Y	Z			Cell data	Tracers	Images
1	0	0	0	7788	58.35	8.700	8.69	8.7
2	6	0	0	7783	57.71	8.700	8.70	8.7
3	–6	0	0	7788	57.72	8.700	8.69	8.7
4	0	6	0	7824	58.68	8.700	8.69	8.7
5	0	–6	0	7790	58.44	8.700	8.70	8.7
6	0	0	6	7787	52.40	8.700	8.70	8.7
7	0	0	–6	7773	50.24	8.700	8.69	8.7
8	6	6	0	7815	58.24	8.700	8.69	8.7
9	6	0	6	7784	52.18	8.700	8.69	8.7
10	0	6	6	7814	52.71	8.700	8.70	8.7
11	6	6	6	7813	52.03	8.700	8.70	8.7

Table 29 ALEGRA translation study results

Run	Translation (cm)			Cycles	Run time (h)	DoP (cm)		
	X	Y	Z			Cell data	Tracers	Images
1	0	0	0	8816	41.13	8.450	8.48	8.5
2	6	0	0	8806	41.28	8.450	8.47	8.5
3	–6	0	0	8802	41.67	8.450	8.48	8.5
4	0	6	0	8814	41.23	8.450	8.47	8.5
5	0	–6	0	8813	42.18	8.450	8.47	8.5
6	0	0	6	8817	40.74	8.450	8.48	8.4
7	0	0	–6	8801	41.70	8.450	8.47	8.4
8	6	6	0	8851	41.89	8.450	8.48	8.5
9	6	0	6	8796	41.27	8.450	8.47	8.4
10	0	6	6	8813	41.19	8.450	8.48	8.4
11	6	6	6	8806	41.82	8.450	8.48	8.4

A simple test of rotational invariance was performed in which the entire model system was rotated about the threat axis by some angle. The 131W threat has perfect cylindrical symmetry, but the target only has 4-fold rotation symmetry about the threat axis. Simulations were run in which the system was rotated in 15° increments from 0° to 90°. To accommodate the rotated geometry, the same expanded domain as the translation study was used. Thus, simulations at 0° are replicates of run 3 from the void buffer study (Tables 14 and 15).

Simulations for both codes are consistent with invariance to rotations, as DoP results remain essentially unchanging with angle (Tables 30 and 31). In CTH, rotating the system by 90° produces a different solution state from the initial system at 0° despite the nominally equivalence of the two. This may be due to roundoff errors in the geometric transformation calculations performed by the code. In ALEGRA, rotating the system 90° produces a solution that is mathematically equivalent to the solution for 0°.

Table 30 CTH axial rotation study results

Run	Angle (°)	Cycles	Run time (h)	DoP (cm)		
				Cell data	Tracers	Images
1	0	7788	58.35	8.700	8.69	8.7
2	15	7784	56.82	8.700	8.70	8.7
3	30	7786	54.95	8.700	8.69	8.7
4	45	7786	54.90	8.700	8.70	8.7
5	60	7787	54.92	8.700	8.70	8.7
6	75	7794	57.11	8.700	8.70	8.7
7	90	7785	58.19	8.700	8.69	8.7

Table 31 ALEGRA axial rotation study results

Run	Angle (°)	Cycles	Run time (h)	DoP (cm)		
				Cell data	Tracers	Images
1	0	8816	41.13	8.450	8.48	8.5
2	15	8850	41.63	8.450	8.48	8.5
3	30	8862	41.63	8.450	8.48	8.5
4	45	8827	41.03	8.450	8.48	8.5
5	60	8873	41.74	8.450	8.48	8.5
6	75	8854	41.87	8.450	8.48	8.5
7	90	8816	42.10	8.450	8.48	8.5

An examination of the effects for more complex rotations was performed by varying the direction of the threat axis within the domain. An even more expanded domain spanning 682 × 682 × 750 cells was required to accommodate rotations of the threat axis about the origin.

Approved for public release; distribution is unlimited.

Tables 32 and 33 show simulation results for CTH and ALEGRA respectively. The orientation of the threat axis is indicated by a vector pointing in the (X, Y, Z) direction. There is a significant decrease in DoP for orientations not aligned with one of the coordinate axes. This is attributable to a well-known directionality effect in domains composed of rectilinear elements. In simple terms, the flow of mass, or advection, is tracked via flux across element boundaries, which are fixed in alignment with the coordinate axes. The algorithms that track advection work well when object motion is oriented primarily along the coordinate axes. They are less accurate for objects moving diagonally, leading to distortion of object geometry that can affect simulation solutions.

Table 32 CTH threat axis direction study results

Run	Threat axis			Cycles	Run time (h)	DoP (cm)		
	X	Y	Z			Cell data	Tracers	Images
1	0	0	1	7820	58.97	8.700	8.69	8.7
2	0	0	–1	7768	53.56	8.700	8.69	8.7
3	1	0	0	7775	58.21	8.700	8.70	8.7
4	0	1	0	7791	58.27	8.700	8.70	8.7
5	1	0	1	7756	59.69	8.318	8.33	8.4
6	0	1	1	7757	59.50	8.318	8.33	8.3
7	1	1	0	7761	59.78	8.318	8.33	8.4
8	1	1	1	8941	67.91	8.389	8.37	8.4
9	1	$\sqrt{\pi}$	π	10241	77.25	8.401	8.39	8.4

Table 33 ALEGRA threat axis direction study results

Run	Threat axis			Cycles	Run time (h)	DoP (cm)		
	X	Y	Z			Cell data	Tracers	Images
1	0	0	1	8818	43.04	8.450	8.47	8.5
2	0	0	–1	8814	43.03	8.450	8.48	8.5
3	1	0	0	8805	43.97	8.450	8.48	8.5
4	0	1	0	8825	43.88	8.450	8.47	8.5
5	1	0	1	8670	42.87	8.379	8.37	8.4
6	0	1	1	8670	42.68	8.379	8.37	8.4
7	1	1	0	8670	42.94	8.379	8.37	8.4
8	1	1	1	9324	45.74	8.389	8.38	8.4
9	1	$\sqrt{\pi}$	π	9654	47.11	8.395	8.37	8.4

While DoP changes significantly for off-axis orientations, run times are not much affected for orientations aligned within one of the principle coordinate planes (one of *x*, *y*, or *z* equal to 0). Off-plane orientations, exemplified by runs 8 and 9, show a marked increase in the number of cycles required to reach completion.

Material advection effects are also examined in the next study. Simulations of the model system are constructed with the threat initially spaced some distance from the target. With no outside forces involved, Newton's first law of motion applies and the threat should move at constant velocity through the domain until impacting the target. Thus, the initial standoff distance should have no influence on the resulting DoP.

Simulations using different initial standoff distances were performed. An elongated domain consisting of 380 × 380 × 842 cells was used to accommodate increased standoff distances. Stop times were increased to compensate for the additional time of flight needed to reach the target. With the threat axis aligned in the positive *z*-direction, adverse effects from advection of the threat through the domain should be minimized. The DoP results (Tables 34 and 35) remain unaffected by increasing standoff distance.

Table 34 CTH threat standoff study results

Run	Standoff (cm)	Stop time (μs)	Cycles	Run time (h)	DoP (cm)		
					Cell data	Tracers	Images
1	0.5	300.0	7784	20.80	8.700	8.69	8.7
2	1.0	303.9	7865	20.27	8.700	8.69	8.7
3	2.0	311.7	8064	20.69	8.700	8.69	8.7
4	5.0	335.2	8662	21.90	8.700	8.70	8.7
5	10.0	374.2	9633	23.84	8.700	8.70	8.7

Table 35 ALEGRA threat standoff study results

Run	Standoff (cm)	Stop time (μs)	Cycles	Run time (h)	DoP (cm)		
					Cell data	Tracers	Images
1	0.5	300.0	8817	29.03	8.450	8.48	8.5
2	1.0	303.9	8912	29.50	8.450	8.48	8.5
3	2.0	311.7	9105	30.13	8.450	8.48	8.5
4	5.0	335.2	9716	31.84	8.450	8.48	8.5
5	10.0	374.2	10726	35.21	8.450	8.48	8.5

The final factor examined in this report is the choice of reference frame. Ballistic simulations typically assume the perspective of a stationary target impacted by a moving threat. Physically, system outcomes are invariant to constant offsets in velocity (for velocities much less than the speed of light—relativistic frames of reference are not accounted for in the simulation codes and are beyond the scope of this work). Thus, as long as the velocity difference between objects is fixed, simulation outcomes should remain unchanged.

A series of simulations were performed in which all velocities were offset in increments of –320 m/s until the threat velocity was 0 m/s. To allow sufficient computational space for motion in the negative *z*-direction, an elongated domain consisting of 380 × 380 × 1148 cells was used. Results (Tables 36 and 37) show that changing the reference frame produces a small change in DoP values. There is a general, but not uniform, increase in DoP with increasing target speed. Additionally, simulations involving a moving target require longer run times than those employing a stationary one.

Table 36 CTH reference frame study results

Run	Velocity (m/s)		Cycles	Run time (h)	DoP (cm)		
	Threat	Target			Cell data	Tracers	Images
1	1280	0	7770	26.16	8.700	8.70	8.7
2	960	–320	7851	26.64	8.700	8.72	8.7
3	640	–640	8182	28.08	8.750	8.73	8.7
4	320	–960	8519	29.22	8.700	8.73	8.7
5	0	–1280	9361	31.96	8.750	8.76	8.7

Table 37 ALEGRA reference frame study results

Run	Velocity (m/s)		Cycles	Run time (h)	DoP (cm)		
	Threat	Target			Cell data	Tracers	Images
1	1280	0	8789	38.79	8.450	8.47	8.5
2	960	–320	8676	40.70	8.500	8.52	8.5
3	640	–640	8596	40.89	8.500	8.54	8.4
4	320	–960	8642	40.94	8.550	8.55	8.5
5	0	–1280	8653	40.77	8.500	8.52	8.5

6. Conclusion

This study examined the sensitivity of simulation outcomes in a simple model ballistic system to variations in fundamental computational parameters. Two different codes were used to emphasize code selection as one source of variation. Both CTH and ALEGRA performed consistently well in simulations of the model system, with DoP results largely unaffected by most parameters studied. Where DoP did vary significantly, it did so for expected reasons.

The influence of computational factors on run time was more pronounced. For individual simulations, changes in run time on the order of a few hours may not have practical implications. However, for projects involving large numbers of simulations, variances in run time can accumulate into a significant computational burden. This study highlights some factors to investigate during the planning stages for such projects.

At a more general level, this study provides a template for the systemic investigation of simulation outcome variance. The model system used in this work was deliberately simple in order to highlight the minimum amount of variance associated with ballistic simulations. More complex systems may exhibit much larger responses to changes in these computational factors. Where it is important to quantify this uncertainty, this work suggests one method for doing do.

The results of this study set the stage for future work. One interesting follow-up would be to repeat these simulations using a shaped-charge jet warhead in place of the 131W. Real shaped-charge devices are notoriously sensitive to small variations in construction and deployment. Simulated devices may prove inherently sensitive as well.

One natural extension of this work would be the investigation of variances in the selection of and parameter values for the materials models. The effects of material models in ballistic simulations have been extensively studied, but the results of this report allow computational factors to be separated from material model effects.

Quantifying the variability inherent in ballistic simulations improves the ability of computational modeling to predict the real-world outcomes of such systems. Effects from factors such as threat yaw and velocity, target obliquity, and geometric variation can be separated from purely computational effects in order to more accurately predict the expected range of performance for ballistic protection technologies. As the demand for quantification of outcome variability in ballistic simulations grows, so too will the need for studies such as this one.

7. References

1. Turing AM. On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society. 1936;42(1):230–265.

2. Smith RC. Uncertainty quantification: theory, implementation, and applications. Philadelphia (PA): Society for Industrial and Applied Mathematics; 2014.

3. Sullivan TJ. Introduction to uncertainty quantification. New York (NY): Springer; 2015.

4. Montgomery DC. Design and analysis of experiments. New York (NY): John Wiley and Sons, Inc.; 2001.

5. CTH. CTH shock physics. Albuquerque (NM): Sandia National Laboratories; c2017 [accessed 2017 Feb 03]. http://www.sandia.gov/CTH/index.html.

6. ALEGRA. ALEGRA…the shock and multiphysics family of codes. Albuquerque (NM): Sandia National Laboratories; 2014 Sep 8 [accessed 2017 Feb 03]. http://www.cs.sandia.gov/ALEGRA/Alegra_Home.html.

7. Zukas JA. Introduction to hydrocodes. Boston (MA); Elsevier; 2004.

8. Sjaardema GD. APREPRO: An algebraic preprocessor for parameterizing finite element analyses. Albuquerque (NM): Sandia National Laboratories; 1992 Dec (updated 1999 Sep). Report No.: SAND92-2291.

9. Benson DJ. Momentum advection on a staggered mesh. Journal of Computational Physics. 1992;100(1):143–162.

10. Gooch WA, Burkins M. A ballistic evaluation of Ti-6Al-4V vs. long rod penetrators. Neuilly-sur-Seine (France): NATO Research and Technology Organization; 2001 May. Report No.: RTO-MP-069(II)-(SM2)-13.

11. MIL-DTL-12560K. Armor plate, steel, wrought, homogeneous (for use in combat-vehicles and for ammunition testing). Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2013.

12. Segletes SB. An analysis on the stability of the Mie-Grüneisen equation of state for describing the behavior of shock-loaded materials. Aberdeen Proving Ground (MD): Army Ballistic Research Laboratory (US); 1991 Mar. Report No.: BRL-TR-3214.

13. Zel'dovich YB, Raizer YP. Physics of shock waves and high-temperature hydrodynamic phenomena. Mineola (NY): Dover Publications, Inc.; 2002.

14. Harris P, Avrami L. Some physics of the Gruneisen parameter. Dover (NJ): Picatinny Arsenal; 1972 Sep. Report No.: 4423.

15. Johnson GR, Cook WH. A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures. The Hague (Netherlands): Proceedings of the 7th International Symposium on Ballistics. 1983;541–547.

16. Group GMX-6. Selected Hugoniots. Los Alamos (NM): Los Alamos National Laboratories; 1969 May. Report No.: LA-4167-MS.

17. Steinberg DJ, Cochran SG, Guinan MW. A constitutive model for metals applicable at high-strain rate. Journal of Applied Physics. 1980;51(3):1498–1504.

18. Marsh SP, editor. LASL shock Hugoniot data. Berkeley (CA): University of California Press; 1980.

19. Haynes WM, editor. CRC handbook of chemistry and physics. 97th ed. Boca Raton (FL): CRC Press/Taylor and Francis; 2017.

20. Kittel C. Introduction to solid state physics. 8th ed. Hoboken (NJ): John Wiley and Sons, Inc.; 2005.

21. Steinberg DJ. Equation of state and strength properties of selected materials. Livermore (CA): Lawrence Livermore National Laboratory; 1991 (Change 1 issued 1996). Report No.: UCRL-MA-106439.

22. Kerley GI, Christian-Frear TL. Sandia equation of state data base: seslan file. Albuquerque (NM): Sandia National Laboratories; 1993 June. Report No.: SAND93-1206.

23. Kerley, GI. Multiphase equation of state for iron. Albuquerque (NM): Sandia National Laboratories; 1993 Feb. Report No.: SAND93-0027.

24. Johnson GR, Holmquist TJ. Test data and computational strength and fracture model constants for 23 materials subjected to large strains, high strain rates, and high temperatures. Los Alamos (NM): Los Alamos National Laboratories; 1989 Jan. Report No.: LA-11463-MS.

25. Meyer HW Jr, Kleponis DS. An analysis of parameters for the Johnson-Cook strength model for 2-in-thick rolled homogenous armor. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2001 June. Report No.: ARL-TR-2528.

26. Kirchhoff G. Ueber das verhältnifs der quercontraction zur längendilatation bei stäben von federhartem stahl. Annalen der Physik und Chemie. 1859;108(11):369–392.

27. Ledbetter HM. Stainless-steel elastic constants at low temperatures. Journal of Applied Physics. 1981;52(3):1587–1589.

28. Crawford DA, Schmitt RG, Butler RJ. Spymaster user's guide, version 6.0. Albuquerque (NM): Sandia National Laboratories; 2016 June.

29. Ayachit U. The ParaView guide, community edition. Clifton Park (NY): Kitware Inc.; 2015.

INTENTIONALLY LEFT BLANK.

Appendix A. Partial CTH Input for Baseline Simulation

This appendix is in its original form, without editorial change.

Approved for public release; distribution is unlimited.

The following is a partial preprocessed input file for the baseline simulation of the model ballistic system in CTH. The Aprepro utility would be used to evaluate the embedded variable assignments and logic statements prior to execution of CTH.

```
*********************************************************************
**************           AAA    RRRRRR      LLL        **************
**************          AAAA    RRRRRRRR    LLL        **************
**************         AA AA    RR     RR   LLL        **************
**************        AA  AA    RR     RR   LLL        **************
**************       AAAAAAA    RRRRRRRR    LLL        **************
**************      AAAAAAAA    RRRRRR      LLL        **************
**************     AA     AA    RR   RR     LLL        **************
**************    AA      AA    RR    RR    LLLLLLLLL  **************
**************   AA       AA    RR     RR   LLLLLLLLL  **************
*********************************************************************
**************       Aprepro Script for CTH Input      **************
**************             Daniel Hornbaker            **************
**************               12 Jan 2016               **************
*********************************************************************
*************************BEGIN USER INPUT****************************
***Title
{TITLE= 'Baseline'}

***Run Parameters
*        Stop Time = { TStop = 300} mms

***Mesh variables
*{  Cell = 0.0500} cm Cell Size
*{ VBufP = 2.0} cm Void Buffer for -z
*{VBufXY = 2.0} cm Void Buffer for x/y
*{ VBufZ = 2.0} cm Void Buffer for +z
*{  XSym = 2} Mesh Symmetry (set 0 for X-symmetry, 2 for full size)
*{  YSym = 2} Mesh Symmetry (set 0 for Y-symmetry, 2 for full mesh)

**************BEGIN PENETRATOR*************
***Penetrator variables
*    Rod Diameter: { DRod =  0.95} cm
*      Rod Length: { LRod = 13.10} cm
*        Velocity: {VZRod = 128000} cm/s
*   Insert Offset: { ORod = -0.50} cm
**************END PENETRATOR*************
**************BEGIN TARGET*************
***Witness variables
*Thickness: {TWit = 15} cm
*   Height: {HWit = 15} cm
*    Width: {WWit = 15} cm
**************END TARGET*************
**************************END USER INPUT*****************************

***Mesh Variables
*Xwidth = {dX=Cell} cm
*Ywidth = {dY=Cell} cm
*Zwidth = {dZ=Cell} cm

{Ifndef(XSym)}
*x-symmetry
*{XNum=ceil(     (VBufXY+WWit/2)/dX)} cells in X
{Else}
*full width
*{XNum=ceil((VBufXY+WWit+VBufXY)/dX)} cells in X
{Endif}
```

```
{Ifndef(YSym)}
*y-symmetry
*{YNum=ceil(     (VBufXY+HWit/2)/dY)} cells in Y
{Else}
*full height
*{YNum=ceil((VBufXY+HWit+VBufXY)/dY)} cells in Y
{Endif}
*{ZNum=ceil((VBufP+LRod-ORod+TWit+VBufZ)/dZ)} cells in Z
*{(XNum*YNum*ZNum)/1e6} million cells total

*XLength = {XNum*dX} cm
*YLength = {YNum*dY} cm
*ZLength = {ZNum*dZ} cm

{Ifndef(XSym)}
*x-symmetry
*Xmin = {Xmin=0} cm
{Else}
*full width
*Xmin = {Xmin=-dX*XNum/2} cm
{Endif}
{Ifndef(YSym)}
*y-symmetry
*Ymin = {Ymin=0} cm
{Else}
*full height
*Ymin = {Ymin=-dY*YNum/2} cm
{Endif}
*Zmin = {Zmin=dZ*floor((ORod-LRod-VBufP)/dZ)} cm

*Xmax = {Xmax = Xmin + XNum*dX} cm
*Ymax = {Ymax = Ymin + YNum*dY} cm
*Zmax = {Zmax = Zmin + ZNum*dZ} cm

*Tmax = {Tmax = TStop*1e-6} s simulated time


***Set Run Parameters
CONTROL
        MMP0          *Mixed Material Pressure model
       TSTOP = {Tmax} *Calculation stop time, default 1e30
        FRAC = 1      *Fracture Logic
        TBAD = 1E30   *Thermodynamic error tolerance
  CKVELOCITY = 0.001  *cm/s; Zero velocity threshold, default 0.001
   DTCOURANT = 0.55   *Stability limit
      DTFRAC = 1      *Fraction of normal timestep for initial cycle
  DTINCREASE = 1.068  *max DT(i)/DT(i-1), default 1.068000433
ENDCONTROL


***Timestep Limits
MINDT
  TIME=0 DTMIN=1e-10
ENDMINDT
MAXDT
  TIME=0 DTMAX=1e-03
ENDMAXDT


***Define objects
DIATOM
***********PENETRATOR*************
  TRANSLATE 0, 0, {ORod}
```

```
    PACKAGE Penetrator
      MATERIAL=1
      ITERATION=5 *Subcells
      MVELOCITY=0,0,{VZRod}
      INSERT R2DP
        CE1=0,0,{-LRod}
        CE2=0,0,0
        P1=0.0000, 0.0000
        P2=0.4850, 0.0000
        P3=0.4850,11.4700
        P4={.485-1.5*tand(7.75)},12.9700
        P5=0.0925,12.9700
        P6=0.0925,13.3400
        P7=0.0000,13.3400
      ENDINSERT
    ENDPACKAGE
  ENDTRANSLATE
************TARGET*************
  TRANSLATE {-WWit/2},{-HWit/2},0
    PACKAGE Witness1
      MATERIAL=2
      ITERATION=5 *Subcells
      INSERT BOX
        P1=0,0,0
        P2={WWit},{HWit},{TWit}
      ENDINSERT
    ENDPACKAGE
  ENDTRANSLATE
ENDDIATOM


***Set the Equation of State for materials
EOS
*Penetrator Material
  MATERIAL1 MGR USER R0=19.235 CS=3.98e5 S=1.24 G0=1.72 CV=0.160e11
*Target Material
  MATERIAL2 MGR USER R0=7.85 CS=4.529e5 S=1.49 G0=1.67 CV=0.518e11
ENDEOS


***Set the Elastic-Plastic and Damage models for materials
EPDATA
  VPSAVE
  MIX 3
*Penetrator Material
  MATEP=1
    JOHNSON-COOK=USER
      ajo=1.507e10   bjo=0.1766e10  cjo=0.016
      mjo=1.00       njo=0.12       tjo=1.4848E-01
      poisson=0.310
*Target Material
  MATEP=2
    JOHNSON-COOK=USER
      ajo=0.780e10   bjo=0.780e10   cjo=0.004
      mjo=1.00       njo=0.106      tjo=1.5365E-01
      poisson=0.294
ENDEPDATA

***Set the Fracture behavior for materials
FRACTURE
  PRESSURE
*Penetrator Material
  PFRAC1=-6.757e10
```

```
*Target Material
  PFRAC2=-2.50e10
  PFMIX=-1e30
  PFVOID=-1e30
ENDFRACTURE


***Material Convection
CONVCT
  CONVECTION=0
  INTERFACE=SMYRA
  NOFRAGMENT=0 *disable fragment moving model for void material
ENDCONVCT


***Material Discard Criteria
DISCARD

ENDDISCARD


***Place Tracers
TRACER
  ADD 0,0,{ORod-dZ} TO 0,0,{ORod-dZ-(18/19)*(LRod-2*dZ)} NUMBER=10
  ADD  0,0,{ORod-dZ-(1/19)*(LRod-2*dZ)}  TO  0,0,{ORod-LRod+dZ}  NUMBER=10
FIX=XY

*Target
  ADD { WWit/2-dX},{ HWit/2-dY},{TWit-dZ} *+x/+y
{Ifndef(XSym)}
  *x-symmetry
  ADD {       +dX},{ HWit/2-dY},{TWit-dZ} *-x/+y
{Ifndef(YSym)}
  *y-symmetry
  ADD { WWit/2-dX},{       +dY},{TWit-dZ} *+x/-y
  ADD {       +dX},{       +dY},{TWit-dZ} *-x/-y
{Else}
  *full height
  ADD { WWit/2-dX},{-HWit/2+dY},{TWit-dZ} *+x/-y
  ADD {       +dX},{-HWit/2+dY},{TWit-dZ} *-x/-y
{Endif}
{Else}
  *full width
  ADD {-WWit/2+dX},{ HWit/2-dY},{TWit-dZ} *-x/+y
{Ifndef(YSym)}
  *y-symmetry
  ADD { WWit/2-dX},{      +dY},{TWit-dZ} *+x/-y
  ADD {-WWit/2+dX},{      +dY},{TWit-dZ} *-x/-y
{Else}
  *full height
  ADD { WWit/2-dX},{-HWit/2+dY},{TWit-dZ} *+x/-y
  ADD {-WWit/2+dX},{-HWit/2+dY},{TWit-dZ} *-x/-y
{Endif}
{Endif}
ENDTRACER
*********************************************************************
```

INTENTIONALLY LEFT BLANK.

Appendix B. Partial ALEGRA Input for Baseline Simulation

This appendix is in its original form, without editorial change

The following is a preprocessed input file for the baseline simulation of the model ballistic system in ALEGRA. The Aprepro utility would be used to evaluate the embedded variable assignments and logic statements prior to execution of ALEGRA.

```
*********************************************************************
**************           AAA    RRRRRR      LLL        **************
**************          AAAA    RRRRRRRR    LLL        **************
**************         AA AA    RR     RR   LLL        **************
**************        AA  AA    RR     RR   LLL        **************
**************       AAAAAAA    RRRRRRRR    LLL        **************
**************      AAAAAAAA    RRRRRR      LLL        **************
**************     AA     AA    RR   RR     LLL        **************
**************    AA      AA    RR    RR    LLLLLLLLL  **************
**************   AA       AA    RR     RR   LLLLLLLLL  **************
*********************************************************************
**************      Aprepro Script for ALEGRA Input    **************
**************             Daniel Hornbaker            **************
**************               12 Jan 2016               **************
*********************************************************************
*************************BEGIN USER INPUT****************************
***Title
*{TITLE= 'Baseline'}

***Run Parameters
*        Stop Time = { TStop = 300} mms

***Mesh variables
*{  Cell = 0.0500} cm Cell Size
*{ VBufP = 2.0} cm Void Buffer for -z
*{VBufXY = 2.0} cm Void Buffer for x/y
*{ VBufZ = 2.0} cm Void Buffer for +z
*{  XSym = 2} Mesh Symmetry (set 0 for X-symmetry, 2 for full size)
*{  YSym = 2} Mesh Symmetry (set 0 for Y-symmetry, 2 for full mesh)


*{eV2K=11604.505} K/eV temperature conversion
**************BEGIN PENETRATOR*************
***Penetrator variables
*    Rod Diameter: { DRod =  0.95} cm
*      Rod Length: { LRod = 13.10} cm
*        Velocity: {VZRod = 128000} cm/s
*   Insert Offset: { ORod = -0.50} cm
**************END PENETRATOR*************
**************BEGIN TARGET*************
***Witness variables
*Thickness: {TWit = 15} cm
*   Height: {HWit = 15} cm
*    Width: {WWit = 15} cm
**************END TARGET*************
**************************END USER INPUT*****************************

UNITS, CGS             *Temp in K

***Mesh Variables
*Xwidth = {dX=Cell} cm
*Ywidth = {dY=Cell} cm
*Zwidth = {dZ=Cell} cm

{Ifndef(XSym)}
*x-symmetry
```

```
*{XNum=ceil(     (VBufXY+WWit/2)/dX)} cells in X
{Else}
*full width
*{XNum=ceil((VBufXY+WWit+VBufXY)/dX)} cells in X
{Endif}
{Ifndef(YSym)}
*y-symmetry
*{YNum=ceil(     (VBufXY+HWit/2)/dY)} cells in Y
{Else}
*full height
*{YNum=ceil((VBufXY+HWit+VBufXY)/dY)} cells in Y
{Endif}
*{ZNum=ceil((VBufP+LRod-ORod+TWit+VBufZ)/dZ)} cells in Z
*{(XNum*YNum*ZNum)/1e6} million cells total

*XLength = {XNum*dX} cm
*YLength = {YNum*dY} cm
*ZLength = {ZNum*dZ} cm

{Ifndef(XSym)}
*x-symmetry
*Xmin = {Xmin=0} cm
{Else}
*full width
*Xmin = {Xmin=-dX*XNum/2} cm
{Endif}
{Ifndef(YSym)}
*y-symmetry
*Ymin = {Ymin=0} cm
{Else}
*full height
*Ymin = {Ymin=-dY*YNum/2} cm
{Endif}
*Zmin = {Zmin=dZ*floor((ORod-LRod-VBufP)/dZ)} cm

*Xmax = {Xmax = Xmin + XNum*dX} cm
*Ymax = {Ymax = Ymin + YNum*dY} cm
*Zmax = {Zmax = Zmin + ZNum*dZ} cm

*Tmax = {Tmax = TStop*1e-6} s simulated time


***Problem Time Control
TERMINATION TIME = {Tmax}


***Specify the Physics
SOLID DYNAMICS
  IGNORE KINEMATIC ERRORS

  ***Timestep Control
  GRADUAL STARTUP FACTOR  1.00
  TIME STEP SCALE         0.90
  MINIMUM TIME STEP       1E-11
  MAXIMUM TIME STEP LIMIT 1E-03
  MAXIMUM TIME STEP RATIO 1.068
  MAXIMUM VOLUME CHANGE   0.5

  ***Domain Control
  DOMAIN
    HIS ADVECTION
    INTERNAL ENERGY ADVECTION
    SMYRA INTERFACE TRACKER
  ***Block Control
  BLOCK 1
    EULERIAN MESH
    ADD DIATOM INPUT
    MODERATE ADVECTION
  END

  ***Multi-material cells
  ISENTROPIC MULTIMATERIAL ALGORITHM
    PRESSURE RELAXATION    = ON
    TEMPERATURE RELAXATION = ON
    THERMAL EQUILIBRIUM    = OFF
  END

  PISCES HOURGLASS CONTROL
    VISCOSITY 0.05
  END

  ***Define objects
  DIOM INSERTION ALGORITHM, HEX
  DIATOM
  ***********PENETRATOR*************
    TRANSLATE 0 0 {ORod}
      PACKAGE Penetrator
        MATERIAL=1
        ITERATION=5 *Subcells = (2^Iter)^Dim
        MVELOCITY= 0.0 0.0 {VZRod}
        INSERT R2DP
          CE1=0 0 {-LRod}
          CE2=0 0 0
          P1=0.0000  0.0000
          P2=0.4850  0.0000
          P3=0.4850 11.4700
          P4={.485-1.5*tand(7.75)} 12.9700
          P5=0.0925 12.9700
          P6=0.0925 13.3400
          P7=0.0000 13.3400
        ENDINSERT
      ENDPACKAGE
    ENDTRANSLATE
  ************TARGET*************
    TRANSLATE {-WWit/2} {-HWit/2} 0
      PACKAGE Witness1
        MATERIAL=2
        ITERATION=5 *Subcells = (2^Iter)^Dim
        INSERT BOX
          P1= 0.0 0.0 0.0
          P2={WWit} {HWit} {TWit}
        ENDINSERT
      ENDPACKAGE
    ENDTRANSLATE
  ENDDIATOM

  ***Cell Doctor
  CELL DOCTOR

  END

  ***Tracers
  TRACER POINTS
*{_i = 0}
{loop(10)}
```

```
*{++_i}
    LAGRANGIAN TRACER {_i} X=0.0 Y=0.0 Z={ORod-dZ-(_i-1)*(LRod-2*dZ)*2/19}
{Endloop}
*{_i = 0}
{loop(10)}
*{++_i}
    LAGRANGIAN TRACER {_i+10} X=0.0 Y=0.0 Z={ORod-dZ-(1/19)*(LRod-2*dZ)-
(_i-1)*(LRod-2*dZ)*2/19}
      CONSTRAIN, X=0, Y=0, Z=1
{Endloop}

    *Target
    LAGRANGIAN TRACER 21  X={ WWit/2-dX} Y={ HWit/2-dY} Z={TWit-dZ}
{Ifndef(XSym)}
    *x-symmetry
    LAGRANGIAN TRACER 22  X={       +dX} Y={ HWit/2-dY} Z={TWit-dZ}
{Ifndef(YSym)}
    *y-symmetry
    LAGRANGIAN TRACER 23  X={ WWit/2-dX} Y={       +dY} Z={TWit-dZ}
    LAGRANGIAN TRACER 24  X={       +dX} Y={       +dY} Z={TWit-dZ}
{Else}
    *full height
    LAGRANGIAN TRACER 23  X={ WWit/2-dX} Y={-HWit/2+dY} Z={TWit-dZ}
    LAGRANGIAN TRACER 24  X={       +dX} Y={-HWit/2+dY} Z={TWit-dZ}
{Endif}
{Else}
    *full width
    LAGRANGIAN TRACER 22  X={-WWit/2+dX} Y={ HWit/2-dY} Z={TWit-dZ}
{Ifndef(YSym)} *y-symmetry
    LAGRANGIAN TRACER 23  X={ WWit/2-dX} Y={       +dY} Z={TWit-dZ}
    LAGRANGIAN TRACER 24  X={-WWit/2+dX} Y={       +dY} Z={TWit-dZ}
{Else}
    *full height
    LAGRANGIAN TRACER 23  X={ WWit/2-dX} Y={-HWit/2+dY} Z={TWit-dZ}
    LAGRANGIAN TRACER 24  X={-WWit/2+dX} Y={-HWit/2+dY} Z={TWit-dZ}
{Endif}
{Endif}
  END
END


***Material Definitions
MATERIAL 1 "PENETRATOR"
  MODEL 101
  INITIAL THERMODYNAMIC STATE, NIST
END
MATERIAL 2 "TARGET"
  MODEL 201
  INITIAL THERMODYNAMIC STATE, NIST
END
***Combined Materials Models
MODEL 101 CTH ELASTIC PLASTIC *WHA
  EOS MODEL 110
  YIELD MODEL 111
  VOID INSERTION MODEL 113
  POISSONS RATIO=0.310
  PHASE CONTROL=1
END

MODEL 201 CTH ELASTIC PLASTIC *RHA
  EOS MODEL 210
  YIELD MODEL 211
  VOID INSERTION MODEL 213
  POISSONS RATIO=0.294
  PHASE CONTROL=1
END


***EOS Models
MODEL 110 KEOS MIEGRUNEISEN *WHA
  R0 = 19.235
  CS = 3.980E5
  CV = {0.160e11/eV2K}
  S1 = 1.24
  G0 = 1.72
  T0 = 293.15 *NIST condition
         MINCS = 0    *sound speed error handling
          CLIP = 1.0  *prevents temp below specified value
  DENSITY CLIP = 0.95 *limits max attainable density
END
MODEL 210 KEOS MIEGRUNEISEN *RHA
  R0 = 7.85
  CS = 4.529E5
  CV = {0.518e11/eV2K}
  S1 = 1.49
  G0 = 1.67
  T0 = 293.15 *NIST condition
         MINCS = 0    *sound speed error handling
          CLIP = 1.0  *prevents temp below specified value
  DENSITY CLIP = 0.95 *limits max attainable density
END


***Yield Models
MODEL 111 JOHNSON COOK EP *WHA
  AJO=1.507e10
  BJO=0.1766e10
  CJO=0.016
  MJO=1.00
  NJO=0.12
  TJO={1.4848E-01*eV2K}
END
MODEL 211 JOHNSON COOK EP *RHA
  AJO=0.780e10
  BJO=0.780e10
  CJO=0.004
  MJO=1.00
  NJO=0.106
  TJO={1.5365E-01*eV2K}
END

***Void Insertion Models
MODEL 113 VOID INSERTION *WHA
  INIT FRAC PRES=-6.757e10
  CYCLES TO FAIL 10
  FORCE FRACTURE 1
  DENSITY THRESHOLD={0.8*19.235}  *minimum allowed material density
END
MODEL 213 VOID INSERTION *RHA
  INIT FRAC PRES=-2.50e10
  CYCLES TO FAIL 10
  FORCE FRACTURE 1
  DENSITY THRESHOLD={0.8*7.85}    *minimum allowed material density
END
*********************************************************************
```

Approved for public release; distribution is unlimited.

List of Symbols, Abbreviations, and Acronyms

3-D	3-dimensional
ARL	US Army Research Laboratory
CGS	centimeter–gram–second
DOD	US Department of Defense
DoP	depth of penetration
EOS	equation of state
RHA	rolled homogenous armor
WHA	tungsten heavy alloy

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO L
IMAL HRA MAIL & RECORDS MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

1 DIR USARL
(PDF) RDRL WMP E
D HORNBAKER